

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

«До захисту допущено»

Завідувач кафедри

(ініціали, прізвище)

“ ” 2020p.

Дипломний проєкт

на здобуття ступеня бакалавра

зі спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: Інформаційна система розпізнавання тексту технології класифікації
додатків до атестатів

Виконав: студент IV курсу, групи ІС-62

Марченков Іван Денисович

(підпис)

Керівник викладач, кандидат наук, доцент, Фіногенов О. Д.

(підпис)

Консультант нормоконтроль доцент, кандидат наук, Новінський В.П.
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент доцент каф. АПЕПС, ТЕФ, к.т.н., Залевська О.В.
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент _____ (підпис)

Київ – 2020 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет Інформатики та обчислювальної техніки

Кафедра Автоматизованих систем обробки інформації і управління

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 122 «Комп'ютерні науки та інформаційні технології»

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

(ініціали, прізвище)

«__» _____ 2020 р.

**ЗАВДАННЯ
на дипломний проєкт студенту**

Марченкову Івану Денисовичу

1. Тема проєкту Інформаційна система розпізнавання тексту
технології класифікації додатків до атестатів,

керівник проєкту Фіногенов Олексій Дмитрович, доцент, кандидат наук,
затверджені наказом по університету від «__» _____ 2020 р. № _____

2. Термін подання студентом проєкту 08.06.2020

3. Вихідні дані до проєкту Модель класифікації зображень оцінок,
написаних рукописом, модель бінарної класифікації додатків до атестатів та
відмінних від нього документів, дані у вигляді зліпка бази даних для оцінки
ефективності реалізованих алгоритмів класифікації.

4. Зміст пояснювальної записки Оптимізувати розрахунок середнього балу
атестату абітурієнта при обробці зображень атестатів під час вступної
кампанії. Розробити модель класифікації зображень оцінок, написаних
рукописом.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) Схема структурна варіантів використання, схема структурна діяльності, схема структурна класів програмного забезпечення, схема структурна компонентів програмного забезпечення, креслення вигляду екранних форм, креслення вигляду звітних форм

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання	завдання
		видав	прийняв
Нормоконтроль	Новінський В. П.		

7. Дата видачі завдання 01.10.2019

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз вимог	10.02.2020	+
2	Побудова алгоритму	24.02.2020	+
3	Опис предметного середовища	23.03.2020	+
4	Створення бази зображень для тренування нейронної мережі	20.04.2020	+
5	Розробка програмного забезпечення	27.04.2020	+
6	Розробка ТЗ	04.05.2020	+
7	Опис функціональної моделі	11.05.2020	+
8	Огляд наявних аналогів	11.05.2020	+
9	Налагодження програми	18.05.2020	+

Студент

(підпис)

І. Д. Марченков

Керівник проєкту

(підпис)

О. Д. Фіногенов

[illegible]

Пояснювальна записка до дипломного проєкту

на тему:

Інформаційна система розпізнавання тексту
технології класифікації додатків до атестатів

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проєкту складається з п'яти розділів, містить 31 рисунок, 7 таблиць, 1 додаток, 24 джерела.

Дипломна бакалаврська робота розглядає проблеми, які виникають при обробці зображень додатків до атестатів абітурієнтів під час вступної кампанії, на прикладі аналізу більше 30 тисяч зображень додатків до атестата, поданих в КПП ім. Ігоря Сікорського в 2018 році. Наведені типові приклади зображень документів і зроблені висновки про можливість автоматичного визначення середнього балу на підставі сканованих копій.

У розділі інформаційного забезпечення вказано, яким чином дані потрапляють до системи, як формується набір даних зображень, як попередньо оброблюється зображення для навчання моделі.

Розділ математичного забезпечення представлений змістовною та математичною постановками задачі для подальшого аналізу способу вирішення завдання, демонструючи детальне застосування математичних методів. У першу чергу проаналізовано можливі методи класифікації зображень та зроблений акцент на застосуванні згорткових нейронних мереж (CNN), які є найсучаснішим алгоритмом класифікації та розпізнавання об'єктів на зображенні. Детально описано метод розв'язання: архітектура нейронної мережі, алгоритми регуляції вагів моделі та типи шарів. для шару виходу використовуватиметься функція *softmax*, для інших шарів – ReLU.

Програмне забезпечення реалізовано на мовою Python. На даний момент Python є найпопулярнішою мовою програмування для досліджень та розробок у машинному навчанні, а для побудови та тренування нейронних мереж застосовано бібліотеку Tensorflow, розробленою компанією Google.

У технологічному розділі вказано, які інструменти повинні бути

					ДП ІС-6212.0000-с.ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

встановлені на ПК для вдалого запуску продукту. Проведено тести, які демонструють коректність роботи програми.

АТЕСТАТ, КЛАСИФІКАЦІЯ ДОКУМЕНТІВ, ОБРОБКА
ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ, НЕЙРОННІ МЕРЕЖІ

					ДП ІС-62 12.0000-с.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

Structure and scope of work. The explanatory note of the graduate project consists of five paragraphs, contains 31 figures, 7 tables, 1 addition, 24 sources.

Thesis bachelor's thesis considers the problems that arise when processing images of appendices to the certificates of entrants during the admission campaign, on the example of the analysis of more than 30 thousand images of appendices to the certificate submitted to Igor Sikorsky Kyiv Polytechnic Institute in 2018. Typical examples of images of documents are given and conclusions about the possibility of automatic determination of the average score on the basis of scanned copies are made.

The paragraph of information support indicates how the data enters the system, how the image data set is formed, how the image is pre-processed to teach the model.

The paragraph of mathematical support is presented by meaningful and mathematical formulations of the problem for further analysis of the method of solving the problem, demonstrating the detailed application of mathematical methods. First of all, the possible methods of image classification are analyzed and the emphasis is placed on the use of convolutional neural networks (CNN), which are the most modern algorithm for classification and recognition of objects in the image. The solution method is described in detail: neural network architecture, model weight control algorithms and layer types. The *softmax* function will be used for the output layer, ReLU for other layers.

The software is implemented in Python. Python is currently the most popular programming language for research and development in machine learning, and the Tensorflow library, developed by Google, has been used to build and train neural networks.

The technology paragraph specifies which tools must be installed on the PC to run the product successfully. Tests were conducted to demonstrate the correct

					ДП IC-6212.0000-с.ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

operation of the program.

CERTIFICATE, DOCUMENT CLASSIFICATION, IMAGE
PROCESSING, MACHINE LEARNING, NEURAL NETWORKS

ЗМІСТ

АНОТАЦІЯ.....	6
ABSTRACT	8
ВСТУП.....	12
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	15
1.1 ОПИС ПРЕДМЕТНОГО СЕРЕДОВИЩА.....	15
1.1.1 Опис предметної області.....	15
1.2 ОПИС ПРОЦЕСУ ДІЯЛЬНОСТІ	23
1.3 ОПИС ФУНКЦІОНАЛЬНОЇ МОДЕЛІ	25
1.4 ОГЛЯД НАЯВНИХ АНАЛОГІВ.....	27
1.5 ПОСТАНОВКА ЗАДАЧІ	30
1.5.1 Призначення розробки	30
1.5.2 Мета розробки.....	30
1.5.3 Задачі розробки.....	31
Висновок до розділу	31
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	33
2.1 ВХІДНІ ДАНІ	33
2.2 ВИХІДНІ ДАНІ.....	34
2.3 СТРУКТУРА МАСИВІВ ІНФОРМАЦІЇ	34
Висновок до розділу	36
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	38
3.1 ЗМІСТОВНА ПОСТАНОВКА ЗАДАЧІ	38
3.2 МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ.....	38
3.3 ОБҐРУНТУВАННЯ МЕТОДУ РОЗВ’ЯЗАННЯ	40
3.3.1 Техніки класифікації зображень.....	41
3.4 ОПИС МЕТОДУ РОЗВ’ЯЗАННЯ	46
3.4.1 Convolutional Neural Network.....	46
3.4.2 Пряме поширення	48
3.4.3 Стохастичний градієнтний спуск (SGD)	49
3.4.4 Шар входу, вихід та позначення та примітки	51
3.4.5 Шар ReLU	52
Висновок до розділу	54
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	56
4.1 ЗАСОБИ РОЗРОБКИ.....	56
4.2 ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ	59

4.3	АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	59
4.3.1	Діаграма класів.....	59
4.3.2	Діаграма послідовностей.....	60
4.3.3	Діаграма компонентів.....	63
4.3.4	Специфікація функцій	64
	Висновок до розділу	68
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	69
5.1	КЕРІВНИЦТВО КОРИСТУВАЧА	69
5.2	ВИПРОБУВАННЯ ПРОГРАМНОГО ПРОДУКТУ	76
5.2.1	Мета випробувань.....	76
5.2.2	Загальні положення	77
5.2.3	Результати випробувань.....	77
	Висновок до розділу	79
	ЗАГАЛЬНІ ВИСНОВКИ.....	81
	ПЕРЕЛІК ПОСИЛАНЬ.....	82
	ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО КОДУ	85
	ФАЙЛ CLASSIFIER.PY.....	85
	ФАЙЛ MARK.PY	88

ВСТУП

Атестат (з 2020 року – свідоцтво) про здобуття повної загальної середньої освіти – документ про освіту державного зразка, який підтверджує отримання повної загальної освіти.

Атестат про здобуття повної загальної середньої освіти видається учням (вихованцям), які закінчили старшу школу 11-й (12-й клас або відповідний курс у професійно-технічному або вищому навчальному закладі I-II рівня акредитації.

Атестат про здобуття повної загальної середньої освіти дає право на вступ до професійно-технічних та вищих навчальних закладів усіх типів і форм власності.

Кожного року при вступі абітурієнтів до вищих навчальних закладів проходить складний розподільчий поділ з рейтинговою пріоритетною системою. Однією зі складових формування рейтингового балу абітурієнта, окрім балів зовнішнього незалежного оцінювання (ЗНО) є середній бал атестату (документу про повну середню освіту), який є чинником впливу на конкурсний бал (в період з 2016 по 2019 роки – до 10% від максимального конкурсного балу 200). Інформація про середній бал вноситься в електронну систему абітурієнтами.

Під час реєстрації в електронному кабінету вступник має обчислити бал атестата самостійно за 12-бальною шкалою з округленням до десятих частин балу і розрахувати як середнє арифметичне усіх зазначених в додатку до атестата оцінок. Додатково вступник завантажує скановану копію (фотокопію) додатка до документа про повну загальну середню освіту.

З 2018 року середні навчальні заклади не мають обов'язкової вимоги підраховувати середній бал та вписувати його в додаток, тому відповідальність за достовірність підрахунку припадає на ВНЗ. Це вимагає

					ДП ІС-6212.0000-с.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

необхідності визначення для великих вищих навчальних закладів порядку 1400-2700 середніх балів атестатів на добу.

Визначення середнього балу атестата абітурієнта як складової його загального рейтингу є однією з трудомістких операцій при обробці заяви, що безпосередньо впливає на кінцевий розподіл абітурієнта. У зв'язку з обмеженим терміном подачі документів (у 2019 році – 13 днів) процедура перевірки достовірності підрахунку середнього балу в додатку до атестату про повну середню освіту і трудомісткістю процедури підрахунку, актуальним є питання про можливість автоматичного розрахунку середнього балу атестата.

У значній частині додатків до атестату поля з балами заповнені рукописом, деякі мають поганий почерк, що ускладнює процес автоматичного розпізнавання.

З вищеописаних проблем можна виділити основні завдання, які повинні бути виконані для автоматичного підрахунку середнього балу додатку до атестату. У роботі ставиться за ціль виконати частину цих завдань.

Першим завданням є класифікація зображення на відповідність для подальшої обробки (додаток до атестату чи сторонній документ, фото тощо).

Другим завданням є автоматичне повертання зображення у правильне положення, якщо воно нахилено під кутом.

Третім завданням є позбавлення зображення від сторонніх елементів. Якщо на зображенні окрім додатку присутні інші документи, фон, треба обрізати зображення таким чином, щоб присутнім був лише додаток.

Четвертим завданням є порогова бінаризація зображення, переведення в чорно-білий формат з кольорового або відтінків сірого.

П'ятим завданням є виділення текстового блоку, в якому знаходяться оцінки за предмети, що розпізнаються, і відкинути інші елементи.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Шостим завданням є сегментація блоків з оцінками на рядки, сегментація рядків на слова та нормалізація тексту.

Сьомим завдання є класифікація зображень з тексом оцінок та підрахунок середнього балу додатку до атестата абітурієнта.

Шосте та сьоме завдання можна виконати іншим способом: за допомогою виявлення усіх зображень оцінок у текстовому блоці. Результат треба записати у csv-файл.

Визначення середнього балу є багатоалгоритмічною задачею, тому завданням проєкту є виконання декількох пунктів алгоритму: бінарна класифікація зображень за коректністю та багатокритеріальна класифікація зображень рукописних оцінок.

Практичне значення одержаних результатів. Розроблено та натреновано модель бінарної класифікації зображень додатків до атестатів та багатокритеріальної класифікації зображень рукописних оцінок.

Публікації. Результати роботи були опубліковані у тезі доповіді «Методи офлайн-розпізнавання рукописного тексту на прикладі задачі автоматичного визначення середнього балу додатку до атестата абітурієнта» на науково-технічній конференції [1].

					ДП ІС-62 12.0000-с.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

1.1.1 Опис предметної області

Під час вступної кампанії абітурієнтів до вищих навчальних закладів проходить складний розподільчий поділ з рейтинговою пріоритетною системою. Однією зі складових формування рейтингового балу абітурієнта, окрім балів зовнішнього незалежного оцінювання (ЗНО) є середній бал атестату (документу про повну середню освіту), який є досить невагомим чинником впливу на конкурсний бал (в період з 2016 по 2019 роки – до 10% від максимального конкурсного балу 200). Інформація про середній бал вноситься в електронну систему абітурієнтами самостійно (треба підрахувати середній бал, якщо цього не зроблено середнім освітнім закладом та завантажити скановану копію додатку до атестату в електронний кабінет).

З 2018 року середні навчальні заклади не мають обов'язкової вимоги підраховувати середній бал та вписувати його в додаток, тому відповідальність за достовірність підрахунку припадає на ВНЗ. Це вимагає необхідності визначення для великих вищих навчальних закладів порядку 1400-2700 середніх балів атестатів на добу (рис. 1.1).

					ДП ІС-6212.0000-с.ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

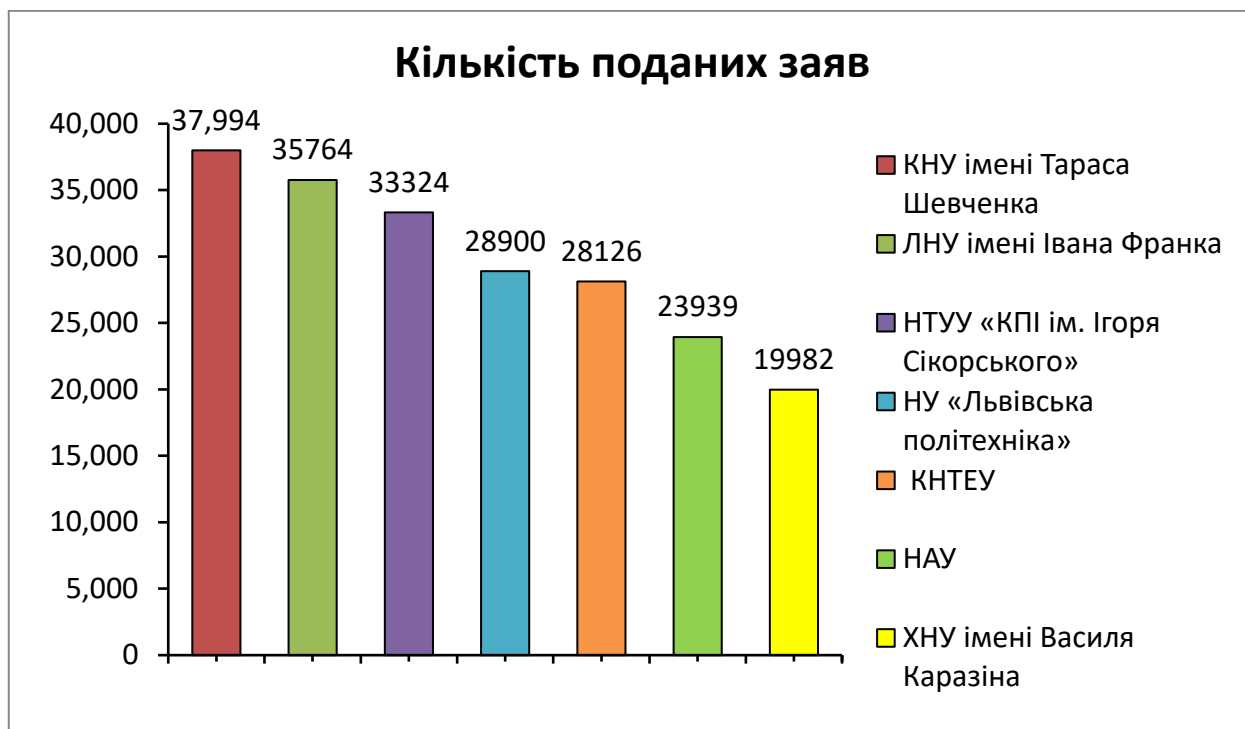


Рисунок 1.1 – Кількість поданих заяв абітурієнтами під час вступної кампанії 2018 року вступаючих на 1-й курс

Відповідно до «Умов прийому на навчання до вищих навчальних закладів України в 2018 році» [2] терміни подачі заяв абітурієнтами, які надходять на основі повної загальної середньої освіти були обмежені датами 12.07.2018 - 26.07.2018. За результатами вступної кампанії від вступників на 1-й курс в 7 ВНЗ було подано близько 20 000 та більше заяв [3]:

- Київський національний університет імені Тараса Шевченка (37 994 заяви);
- Львівський національний університет імені Івана Франка (35764);
- Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» (33324);
- Національний університет «Львівська політехніка» (28900);
- Київський національний торговельно-економічний університет (28126);
- Національний авіаційний університет (23939);
- Харківський національний університет імені Василя Каразіна (19982).

Визначення середнього балу атестата абітурієнта як складової його загального рейтингу є однією з трудомістких операцій при обробці заяви, що безпосередньо впливає на кінцевий розподіл абітурієнта. У зв'язку з обмеженим терміном подачі документів (у 2019 році – 13 днів) процедура перевірки достовірності підрахунку середнього балу в додатку до атестату про повну середню освіту і трудомісткістю процедури підрахунку, актуальним є питання про можливість автоматичного розрахунку середнього балу атестата.

Додаток до атестата є уніфікованим документом, але при аналізі зображень був визначений набір проблем [4], які ускладнюють можливість його обробки, а саме:

1) ряд абітурієнтів завантажують не додаток, а відмінні від нього документи: власне фото, сам атестат або інші документи (рис. 1.2 - 1.3);

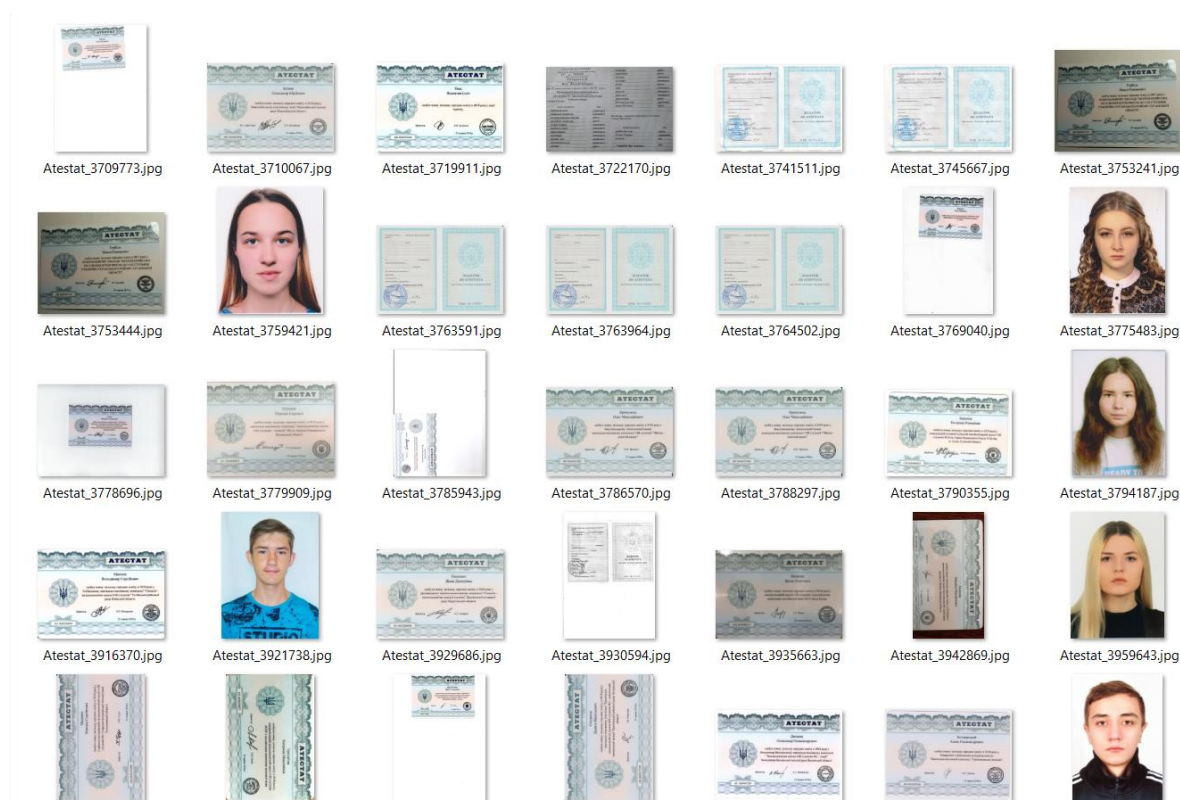


Рисунок 1.2 – Приклади некоректних файлів

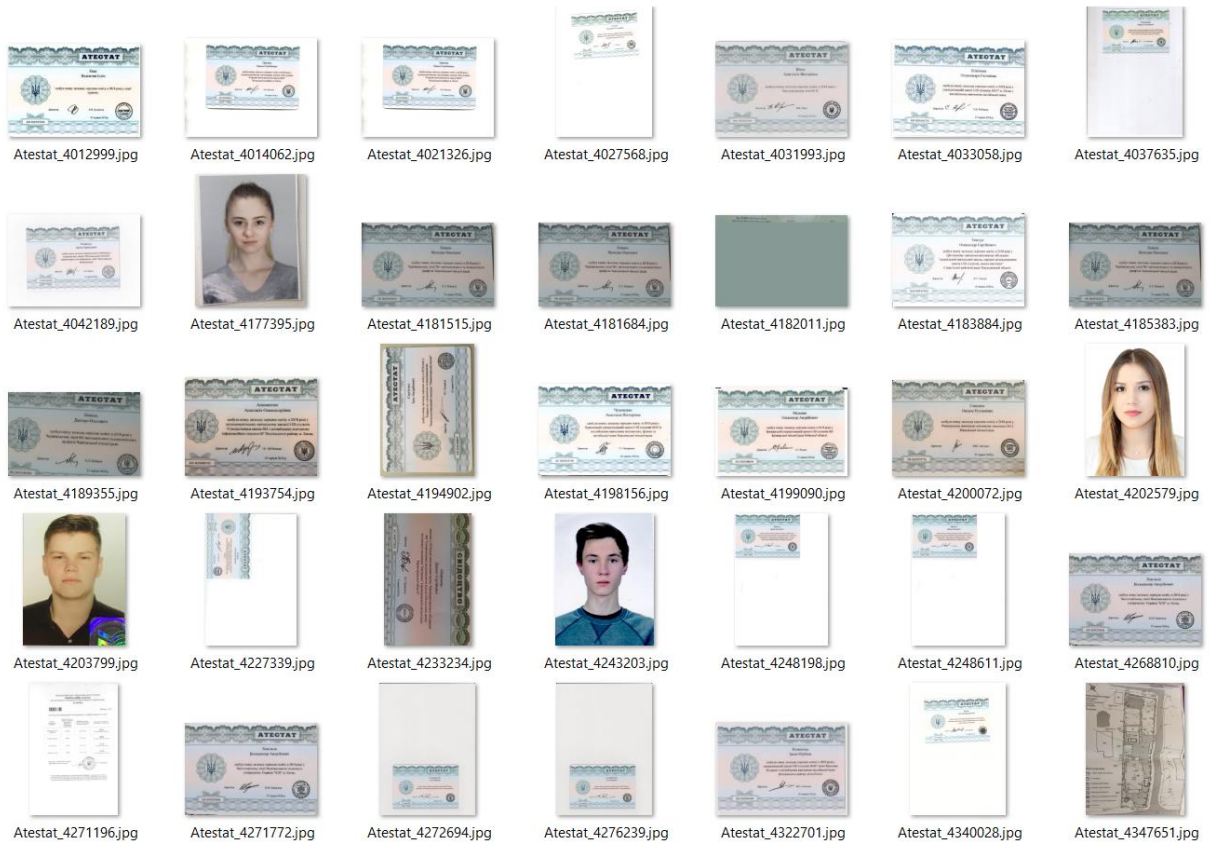


Рисунок 1.3 – Приклади некоректних зображень

2) фонове обрамлення – багато хто з завантажених зображень отримані шляхом не сканування документа, а фотографування (рис. 1.4);



Atestat_3708039.jpg



Atestat_3708207.jpg



Atestat_3708542.jpg



Atestat_3709093.jpg



Atestat_3716770.jpg



Atestat_3725448.jpg



Atestat_3727623.jpg



Atestat_3736591.jpg



Atestat_3741527.jpg



Atestat_3743935.jpg



Atestat_3753015.jpg



Atestat_3753385.jpg



Atestat_3756874.jpg



Atestat_3757574.jpg



Atestat_3782410.jpg

Рисунок 1.4 – Приклади зображень з фоновим обрамленням

3) розташування декількох документів на одному зображенні – часто разом з додатком до атестату (зворотного боку, на якій розташована інформація про ПІБ, школі і оцінках) на зображенні присутній і лицьова сторона і / або сам пластиковий атестат (рис. 1.5);



Atestat_3738854.jpg



Atestat_3720623.jpg



Atestat_3714027.jpg



Atestat_3706609.jpg

Рисунок 1.5 – Приклади скан-копій з декількома документами

4) розташування на зображенні – зустрічаються як представлені в книжковій, так і в альбомній орієнтації. У разі, коли зображення отримані за допомогою фотоапарата або мобільного телефону, документи можуть бути розташовані під різними кутами і мати опуклість в місці перегину. Крім того є зображення, які розташовані від «правильного» положення під кутами $\pm 90^\circ$ або 180° . Окремі додатки мали на оригіналі загини або механічні пошкодження (рис. 1.6);

					ДП ІС-6212.0000-с.ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		



Atestat_3718983.jpg



Atestat_3746541.jpg



Atestat_3755176.jpg



Atestat_3774951.jpg



Atestat_3783302.jpg



Atestat_3783839.jpg



Atestat_3797966.jpg



Atestat_3820600.jpg



Atestat_3895022.jpg



Atestat_4259025.jpg



Atestat_4276869.jpg



Atestat_4360789.jpg

Рисунок 1.6 – Приклади зображень додатків з різними відтінками

- 5) колір – на відміну від сканованих копій, де колір додатки до атестата близький до ціан, в залежності від освітлення зйомки, зустрічаються зображення з кольорами близькими до жовтого або червоного. Крім того, значна частина зображень надана в чорно-білому кольорі (рис. 1.6);
- 6) тіні – при зйомці на великій кількості фотографій є тіні або відблиски фотоспалаху (рис. 1.7);

Змн.	Арк.	№ докум.	Підпис	Дата

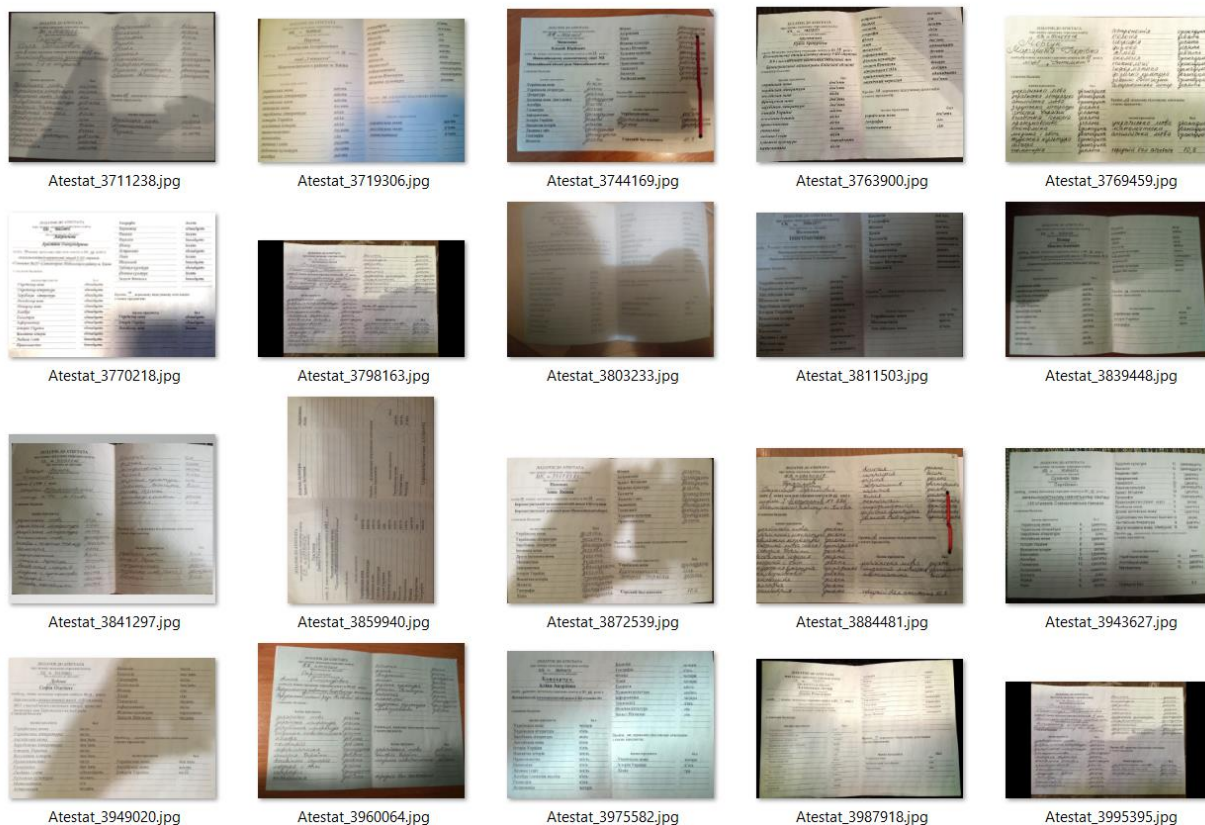


Рисунок 1.7 – Приклади зображень додатків з наявністю тіней

7) якість зображень – в кабінеті вступника присутні обмеження на розмір файлу (менше 1 Мб), але відсутні обмеження і контроль розмірів зображень. Серед аналізованих зображень розміри коливаються від 144x100 до 10200x6650 пікселів;

8) тип файлу – хоча всі додатки розташовані в Єдиній державній електронній базі з питань освіти завантажуються в форматі JPG, але деякі вступники розміщують файли в форматі PDF, замінивши розширення файлу, наприклад, щоб розмістити в одному файлі на декількох сторінках обидві сторони додатку до атестата.

У значній частині додатків до атестату поля з балами заповнені рукописом, деякі мають поганий почерк, що ускладнює процес автоматичного розпізнавання.

З вищеописаних проблем можна виділити основні завдання, які повинні бути виконані для автоматичного підрахунку середнього балу додатку до атестату.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

У роботі ставиться за ціль виконати частину цих завдань.

1.2 Опис процесу діяльності

Запропонований програмний продукт присвячений роботі із зображеннями додатків до атестата. Опис процесу діяльності представлено у вигляді трьох незалежних процесів.

Перший процес – бінарна класифікація зображень за коректністю змісту – додаток або сторонній документ, фото тощо (рис. 1.8).

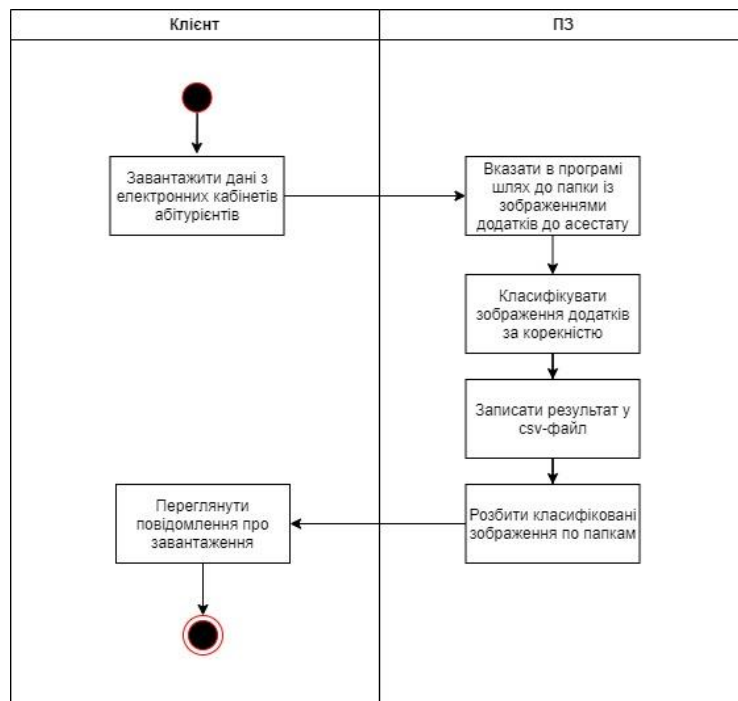


Рисунок 1.8 – Діаграма діяльності класифікації додатків за коректністю

Другий процес – багатокритеріальна класифікація зображень за орієнтацією - 0°, 90°, 180°, 270°. Після цього програма зберігає зображення у правильну орієнтацію (рис. 1.9).

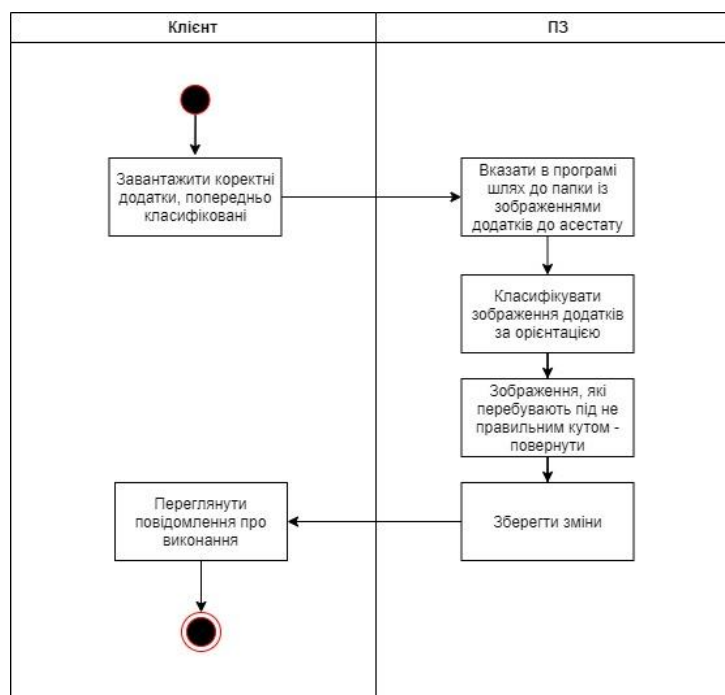


Рисунок 1.9 – Діаграма діяльності класифікації додатків за орієнтацією

Третій процес – класифікація зображень оцінок, написаних рукописом, розбиття їх по папкам та збереження результату у csv-файл.

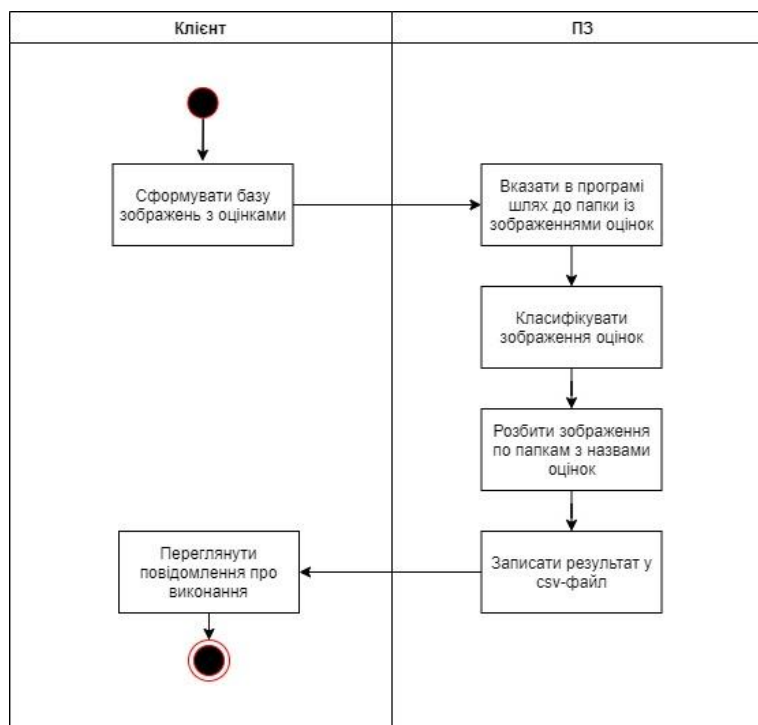


Рисунок 1.10 – Діаграма діяльності класифікації зображень оцінок

1.3 Опис функціональної моделі

Система передбачає наступні варіанти використання:

Таблиця 1.1 – Варіант використання UC001

Назва	Класифікація додатків за коректністю
Опис	Система класифікує зображення додатків до атестату на 2 класи – коректне/некоректне та формує результат, зберігаючи зображення до різних папок.
Учасник	Користувач
Передумови	Задано адресу директорії з базою зображень, формат зображень – JPEG
Постумови	Система розділила файли за категоріями та записала результат у csv-файл
Основний сценарій	Користувач запускає модуль програми. Програма за допомогою завчасно навченої моделі, класифікує зображення. Модель побудовано, за допомогою згорткової нейронної мережі (CNN).

Таблиця 1.2 – Варіант використання UC002

Назва	Повертання зображення у правильну орієнтацію
Опис	Система класифікує зображення додатків до атестату на 4 класи – 0°, 90°, 180°, 270° та повертає зображення у правильне положення.
Учасник	Користувач

Продовження таблиці 1.2

Передумови	Задано адресу директорії з базою зображень, формат зображень – JPEG
Постумови	Система повертає та зберігає зображення у правильному положенні.
Основний сценарій	Користувач запускає модуль програми. Програма за допомогою завчасно навченої моделі, класифікує зображення за орієнтацією. Модель побудовано, за допомогою згорткової нейронної мережі (CNN).

Таблиця 1.3 – Варіант використання UC003

Назва	Класифікація зображень оцінок
Опис	Система класифікує зображення додатків до атестату на 10 класів (оцінки від 3 до 12) та повертає результат в цифровому вигляді.
Передумови	Сформовано обрізані зображення оцінок та вказано адресу директорії. Формат зображень – JPEG
Постумови	Система повертає результат в цифровому вигляді та записує у csv-файл.
Основний сценарій	Користувач запускає модуль програми. Програма за допомогою завчасно навченої моделі, класифікує зображення оцінок. Модель побудовано, за допомогою згорткової нейронної мережі (CNN).

1.4 Огляд наявних аналогів

Google Lens - це технологія розпізнавання зображень, розроблена Google, призначена для отримання відповідної інформації, пов'язаної з об'єктами, які вона ідентифікує за допомогою візуального аналізу на основі нейронної мережі. [5]. Спочатку була представлена як окремий додаток, а потім була інтегрована у стандартний додаток для камер Android.

Google Lens дозволяє копіювати текст у вигляді телефонних номерів, дат, адрес та електронних адрес, а також тексту з реального світу. Інструмент дозволяє розпізнавати номер банківської карти при направленні об'єктиву. Також можна копіювати і вставляти рукописні нотатки з телефону на комп'ютер за допомогою об'єктива (рис. 1.11), хоча це працює лише в тому випадку, якщо ваш почерк достатньо акуратний. Ще однією функцією є інструмент вимови. Достатньо виділити слово в об'єктиві та натиснути «Слухати», щоб почути, як воно вимовляється.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.11 – Офлайн-розпізнавання рукописного тексту за допомогою Google Lens

Aipoly Vision - корисний додаток для розпізнавання об'єктів і кольорів, який допомагає сліпим, із вадами зору та людям, які не розрізняють кольори, зрозуміти своє оточення [6]. Це робиться за допомогою штучного інтелекту для розпізнавання об'єктів через камеру пристрою, після чого користувачу повідомляється назва кожного об'єкта. Цей додаток є прекрасним прикладом того, як нові технології можуть позитивно змінити користувачів зараз, і він оснащений функцією "розумний факел", яка автоматично включає факел пристрою, якщо рамка камери занадто темна, що дозволяє додатку працювати в умовах слабкої освітленості.

Leafsnap – серія електронних посібників, що розробляються

					ДП ІС-6212.0000-с.ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

дослідниками Колумбійського університету, Університету Меріленда та Смітсонівської інституції [7]. Безкоштовні мобільні додатки використовують програмне забезпечення для візуального розпізнавання, щоб допомогти виявити види дерев за фотографіями їх листя. Вони містять красиві зображення з високою роздільною здатністю листя, квітів, плодів, черешків, насіння та кори.

Pen to Print - перший рукописний додаток OCR для перетворення сканованих рукописних текстів у цифровий текст, доступних для редагування, пошуку та зберігання на будь-якій цифровій платформі [8].

Цей сканер OCR використовується для сканування, розпізнавання та перетворення рукописних документів у цифровий текст, який можна редагувати, шукати та зберігати на будь-якому пристрої чи хмарній службі.

Рукописні нотатки, такі як листи, шкільні конспекти, щоденники, протоколи зустрічей, списки продуктів, рецепти тощо, тепер можуть бути скановані та перетворені механізмом розпізнавання рукописних текстів у текст, доступний для використання на будь-якій цифровій платформі.

ABBYY FineReader – програма для оптичного розпізнавання символів, розроблена російською компанією ABBYY.

Програма дозволяє переводити зображення документів (фотографій, сканованих, PDF-файлів) в електронні редаговані формати. Зокрема, в Microsoft Word, Microsoft Excel, Microsoft Powerpoint, Rich Text Format, HTML, PDF / A, searchable PDF, CSV і текстові (plain text) файли. Програма підтримує розпізнавання тексту на 192 мовах і має вбудовану перевірку орфографії для 48 з них [9].

Програма доступна для Windows і macOS. Ядро FineReader без графічного інтерфейсу є для Linux. [10] В основі FineReader технологія

					ДП ІС-6212.0000-с.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

оптичного розпізнавання символів ABBYY OCR, яку ліцензують Fujitsu, Panasonic, Xerox, Samsung [11] та інші.

1.5 Постановка задачі

1.5.1 Призначення розробки

Програмний продукт призначений для облегшення роботи Приймальної комісії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» під час вступної кампанії, шляхом автоматизації обробки даних. Однією зі складових формування рейтингового балу абітурієнта, окрім балів зовнішнього незалежного оцінювання (ЗНО) є середній бал додатку до атестату (документу про повну середню освіту). Інформаційна система повинна проводити обробку зображень вивантажених з електронних кабінетів вступників, а саме підрахунок середнього балу додатку та записувати результат роботи у csv-файл.

1.5.2 Мета розробки

Розглянути проблеми, які виникають під час рукописного офлайн-розпізнавання тексту та розібрати один із способів вирішення цієї задачі. Зменшити витрати часу, запобігти можливим помилкам при підрахунку середнього балу додатку до атестата, покращити продуктивність роботи Приймальної комісії Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» під час вступної кампанії за рахунок автоматизації процесу обчислення середнього балу додатку.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

1.5.3 Задачі розробки

- класифікувати зображення на відповідність для подальшої обробки (додаток до атестату чи стороннє зображення);
- якщо зображення нахилено під кутом, повернути в горизонтальне положення;
- якщо на зображенні окрім додатку присутні сторонні елементи, обрізати зображення таким чином, щоб присутнім був лише додаток;
- переведення зображення в чорно-білий формат з кольорового або відтінків сірого (порогова бінаризація);
- виділити безпосередньо область, в якій знаходяться бали за предмети, що розпізнаються, і відкинути інші елементи;
- розділити область з балами на окремі рядки (сегментувати рядки) і поділити рядки на слова (сегментувати слова), нормалізувати текст;
- розпізнати слова, виокремити числа та підрахувати середнє значення.

Висновок до розділу

Розділ ознайомлює читача з основними нюансами роботи. Робота стосується технології класифікації зображень. У випадку дипломного проєкту, основною ціллю є полегшення роботи Приймальної комісії КПП ім. Ігоря Сікорського під час вступної кампанії, шляхом автоматизації обробки даних.

Під час кампанії формується рейтинг, за допомогою вагової системи впливу критеріїв, таких як, бали ЗНО, середній бал сертифікату (додатку до атестату) про повну середню освіту, конкурсні та додаткові бали. Завданням проєкту є автоматизація обробки додатку до атестату – документу, в якому містяться підсумкові оцінки за період навчання у закладі та бали ЗНО. Результат, який система повинна дати на виході – середній бал додатку.

					ДП ІС-62 12.0000-с.ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

З 2018 року середні навчальні заклади не мають обов'язкової вимоги підраховувати середній бал та вписувати його в додаток, тому відповідальність за достовірність підрахунку припадає на ВНЗ. Через обмежений час обробки даних, вищі, особливо, які входять до топ-рейтингів, повинні розглядати велику кількість заяв абітурієнтів. Наприклад, КПП ім. Ігоря Сікорського під час вступної кампанії повинен оброблювати 1400-2700 додатків за день.

Обробка додатку до атестату є комплексною задачею. У розділі описано, з якими основними проблемами може стикнутися система під час обробки. Зокрема, вступник може завантажити не той документ, надіслати додаток в неправильній орієнтації, зробити фото замість сканування тощо.

Для вирішення цих проблем було сформульовано задачі розробки. Було вирішено, що система буде класифікувати зображення, за допомогою моделі згорткової нейронної мережі (CNN), що є одним з найефективніших рішень у наш час.

На сьогоднішній день існує безліч застосунків для класифікації зображень, наприклад «Google Lens», «Aipoly Vision» тощо. Але для вирішення вузькопрофільної задачі ефективніше буде побудувати власну модель, яка скрадатиметься з обмеженої кількості даних для навчання класифікатору.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Для розроблюваної інформаційної системи дані надходять від користувача.

Вхідними даними даної системи є:

- дані, які надходять від користувача;
- дані, які необхідні для того, щоб система надавала відповідь на запит користувача;
- дані, які необхідні для того, щоб система могла розпізнати зображення, яке завантажив користувач.

Користувач у програмі вказує адреси директорій, в яких знаходяться зображення атестатів та зображення оцінок. Також, система фіксує номер атестату, який присутній у назві файлу зображення.

В інформаційній системі є два варіанти вхідних даних:

- зображення додатків до атестату у форматі JPEG та розміром не більше 1 Мб;
- зображення оцінок, написаних рукописом JPEG та розміром не більше 1 Мб.

Для навчання першої моделі нейронної мережі буде використовуватися набір даних, розділений на 2 класи: правильні/неправильні. Наразі набір даних складається з 360 зображень.

Для навчання другої моделі нейронної мережі буде використовуватися набір даних, розділений на 10 класів: три, чотири, п'ять, шість, сім, вісім, дев'ять, десять, одинадцять, дванадцять. Оцінок 1 та 2 у базі додатків до атестату знайдено не було.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2 Вихідні дані

Результатами роботи першого модулю інформаційної системи є вихідні дані:

- кількість додатків віднесених до коректних;
- кількість додатків віднесених до некоректних;
- зображення за результатами програми розподіляються по папкам «correct» та «incorrect»;
- csv-файл, в якому присутні номер атестату та результат класифікації.

Результатами роботи другого модулю інформаційної системи є вихідні дані:

- кількість додатків віднесених до кожного класу;
- зображення за результатами програми розподіляються по папкам «three», «four», «five», «six», «seven», «eight», «nine», «ten», «eleven» та «twelve»;
- csv-файл, в якому присутні номер атестату та результат класифікації.

2.3 Структура масивів інформації

Для навчання нейронної мережі будуть використовуватися набори даних `certificates_dataset` та `marks_dataset` (рис. 2.1).

Набір даних `certificates_dataset` містить 500 зображень. При навчанні розміри зображень зменшені до 20×20 пікселів. При такій якості модель показує найкращий результат.

Папка «correct» містить 400 зображень. Папка «incorrect» містить 100 зображень. Кожне зображення має назву *Atestat_N*, де *N* – ідентифікатор атестату.

Набір даних `mark_dataset` містить 477 зображень. При навчанні моделі

					ДП ІС-6212.0000-с.ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

розміри зображень зменшені до 36 × 36 пікселів.

Папки містять таку кількість зображень (рис. 2.2):

- «three» містить 1 зображення;
- «four» містить 14 зображень;
- «five» містить 19 зображень;
- «six» містить 28 зображень;
- «seven» містить 51 зображення;
- «eight» містить 56 зображень;
- «nine» містить 69 зображень;
- «ten» містить 89 зображень;
- «eleven» містить 87 зображень;
- «twelve» містить 63 зображень.

Кожне зображення має назву *mark_n_m*, де *n* – ідентифікатор атестату, *m* – оцінка.

```

C:\WINDOWS\system32\cmd.exe
(venv) C:\Users\Admin\Desktop\KPI\Graduate work\Program>tree
Структура папок
Серийный номер тома: 5A79-051F
C:.\
├── certificates_dataset
│   ├── test_dataset
│   │   ├── correct
│   │   └── incorrect
│   └── train_dataset
│       ├── annotation
│       ├── correct
│       └── incorrect
└── marks_dataset
    ├── test_dataset
    │   ├── eight
    │   ├── eleven
    │   ├── five
    │   ├── four
    │   ├── nine
    │   ├── seven
    │   ├── six
    │   ├── ten
    │   └── twelve
    └── train_dataset
        ├── eight
        ├── eleven
        ├── five
        ├── four
        ├── nine
        ├── seven
        ├── six
        ├── ten
        └── twelve
  
```

Рисунок 2.1 – Структура папок

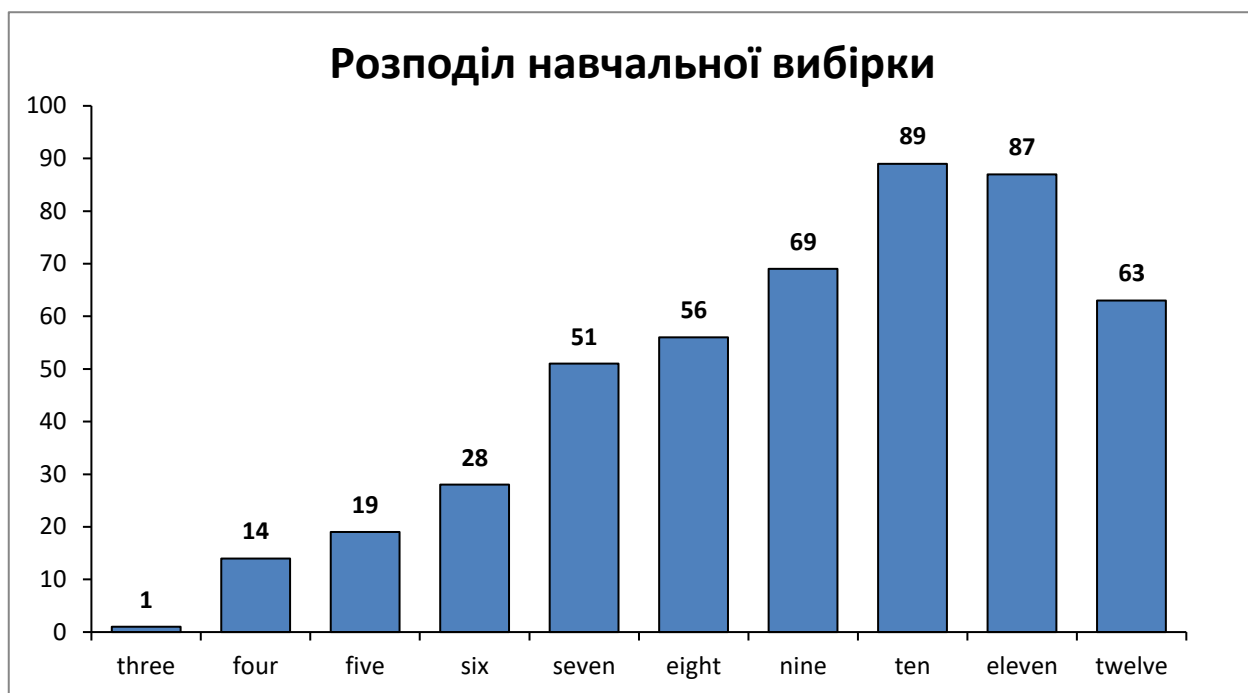


Рисунок 2.2 – Розподіл навчальної вибірки

Висновок до розділу

Для розроблюваної інформаційної системи дані надходять від користувача. Користувач у програмі вказує адреси директорій, в яких знаходяться зображення атестатів та зображення оцінок. Також, система фіксує номер атестату, який присутній у назві файлу зображення.

Для навчання першої моделі нейронної мережі буде використовуватися набір даних, розділений на 2 класи: правильні/неправильні. Наразі набір даних складається з 360 зображень.

Для навчання другої моделі нейронної мережі буде використовуватися набір даних, розділений на 10 класів: три, чотири, п'ять, шість, сім, вісім, дев'ять, десять, одинадцять, дванадцять. Оцінок 1 та 2 у базі додатків до атестату знайдено не було.

Набір даних `certificates_dataset` містить 500 зображень. При навчанні розміри зображень зменшені до 20×20 пікселів.

Набір даних mark_dataset містить 477 зображень. При навчанні моделі розміри зображень зменшені до 36×36 пікселів. При зменшенні зображень до таких розмірів моделі показують найкращий результат.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Програмний продукт призначений для облегшення роботи Приймальної комісії КПП ім. Ігоря Сікорського під час вступної кампанії, шляхом автоматизації обробки даних. Однією зі складових формування рейтингового балу абітурієнта, окрім балів зовнішнього незалежного оцінювання (ЗНО) є середній бал додатку до атестату (документу про повну середню освіту). Інформаційна система повинна проводити обробку зображень вивантажених з електронних кабінетів вступників, а саме підрахунок середнього балу додатку та записувати результат роботи у csv-файл.

3.2 Математична постановка задачі

Розглянемо теорію навчання нейронних мереж в контексті навчання з учителем на основі результатів, опублікованих в [12, 13]. Модель навчання з учителем складається з трьох взаємопов'язаних компонентів, які в математичних термінах описуються наступним чином:

- *середовище*: характеризується розподілом ймовірностей $P_X(x)$ з випадковими і незалежними випадками x ;
- *відомий результат*: співставляє бажаний результат d для кожного із вхідних векторів x , отриманих із зовнішнього середовища, відповідно до умовної функції розподілу $P_X(d|x)$. Ні множина вхідних величин $P_X(x)$, ні правило класифікації $P_X(d|x)$ невідомі. Однак відомо, що існує спільний розподіл ймовірності:

$$P_X(d, x) = P(x) \cdot P_X(d|x). \quad (3.1)$$

Бажаний результат d і вектор вхідних даних x пов'язані наступним співвідношенням

					ДП ІС-6212.0000-с.ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

$$d = f(x, v), \quad (3.2)$$

де v - шум, тобто передбачена похибка в даних, що використовуються для навчання;

– *Модель машинного навчання*: нейронна мережа здатна реалізувати безліч функцій відображення [12], описуваних співвідношенням

$$y = F(x, w), \quad (3.3)$$

де y – результат класифікації, згенерований моделлю, що побудована у відповідь на вхідний вектор x ; w - набір вільних параметрів (синаптичних вагів), які регулюються під час навчання, вибраних з простору параметрів W .

Задача навчання із вчителем [14] полягає у виборі конкретної функції $F(x, w)$, яка оптимально змоделює очікуваний результат d . Вибір, в свою чергу, ґрунтується на N незалежних прикладах навчання:

$$T = \{(d_i, x_i)\}_{i=1}^N \quad (3.4)$$

Кожна пара обирається навчальною моделлю з множини T з деякою узагальненою функцією розподілу ймовірності $P_{X,D}(d|x)$, яка, як і інші функції розподілу, фіксована, але невідома.

Розглянемо задачу навчання із вчителем як задачу апроксимації, що складається з пошуку функції $F(x, w)$, яка найкращим чином моделює бажану функцію $f(x)$.

Нехай $L(d, F(x, w))$ - міра витрат між бажаним результатом d , відповідним вхідному вектору x , і результатом $F(x, w)$, що повернула модель. Тоді очікувана величина витрат визначається функцією середнього ризику:

$$R(w) = \int L(d, F(x, w)) dP_{X,D}(d|x) \quad (3.5)$$

якщо невідома функція розподілу $P(z)$, але дана випадкова вибірка z_1, \dots, z_l .

Особливість полягає в тому, що на функцію $Q(z, \alpha)$ накладено обмеження:

- вектор z задається $n + 1$ координатами: координатою d і координатами x_1, \dots, x_n ;
- функція витрат $Q(z, \alpha)$ задана у вигляді $(d - F(x, \alpha))^2$, де $F(x, \alpha)$ – прогнозований результат класифікації.

Традиційно виділяють три способи вирішення завдання мінімізації функціонала середнього ризику:

- відновлення розподілу ймовірностей;
- побудова рекурентної процедури;
- заміна на функціонал емпіричного ризику.

Кожен із цих способів має свої переваги і недоліки, теорія адаптації розпізнавання образів згодом стала розвиватися у всіх трьох напрямках.

3.3 Обґрунтування методу розв'язання

Цифрове зображення складається з дискретних елементів зображення, які називаються пікселями. Пікселі виражені цифровими числом (digital number, DN), що зображує середнє значення ділянки порівняно невеликої площі із повним розміром. Діапазон значень DN зазвичай становить від 0 до 255. Цифрова обробка зображень - це колекційна техніка для управління цифровими зображеннями комп'ютерами. Класифікація зазвичай складається з 4 кроків:

1) Попередня обробка, напр. атмосферна корекція (atmospheric correction), придушення шуму (noise suppression) та пошук співвідношення смуг (band ratio), аналіз основних компонентів (principal component analysis) тощо;

					ДП ІС-62 12.0000-с.ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

- 2) Тренування: вибір ознак, які найкраще описують модель;
- 3) Рішення: Вибір відповідного методу для порівняння моделей зображення з цільовими;
- 4) Оцінка точності класифікації. Класифікація інформаційних даних системами «з вчителем» та «без вчителя».

3.3.1 Техніки класифікації зображень

Класифікація зображень є важливим завданням для багатьох аспектів досліджень глобальних змін та екологічних застосувань. Розроблено безліч алгоритмів класифікації - від класифікатора максимальної ймовірності до класифікаторів нейронної мережі. У цьому пункті робиться акцент на аналізі та використанні різних передових методів класифікації, таких як Штучні Нейронні Мережі (Artificial Neural Networks), Метод Опорних Векторів (Support Vector Machines), Нечіткі Межі (Fuzzy Measures), Генетичні алгоритми та їх комбінація для класифікації цифрових зображень [15].

3.3.1.1 Штучна нейронна мережа (ANN)

ANN — розподілена обчислювальна система [16], яка має природну концепцію до зберігання досвіду. Вона може надати відповідні рішення для задач, які, як правило, характеризуються нелінійними зв'язками, достатньо високими розмірами, складними, неточними та недосконалими даними, схильних до помилок або відсутністю чітко заявленого математичного рішення чи алгоритму. Ключовою перевагою нейронних мереж є те, що модель системи може бути побудована з наявних даних. Класифікація зображень за допомогою нейронних мереж проводиться шляхом вилучення особливостей текстури, а потім із застосуванням алгоритму зворотного поширення помилки (back propagation algorithm, ВКР).

Навчання комбінованої мережі за допомогою ВКР

Далі передбачається сигмоподібна функція $f(x)$:

					ДП ІС-62 12.0000-с.ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

$$f(x) = \frac{1}{1+e^{-x}} \quad (3.7)$$

Алгоритм зворотного поширення помилки зображений на рис. 3.1.

ВКР реалізується за допомогою наступних кроків:

- 1) Ініціалізація вагів малими випадковими значеннями;
- 2) Проведення вхідних векторів X_0, X_1, \dots, X_6 через мережу, обчислення вагової суми вузла та застосування сигмоподібної функції (sigmoid function). Також, встановлення усіх бажаних виходів d_0, d_1, \dots, d_5 , як правило, до нуля, за винятком класу, з якого подається вхід.

- 3) Обчислити значення помилки для кожного вихідного вузлу за формулою:

$$\delta_i = y_i(1 - y_i)(d_j - y_j), \quad (3.8)$$

де d_j - бажаний вихід вузла j ; і y_j - фактичний вихід.

- 4) Підрахувати значення помилки кожного прихованого вузлу:

$$\delta_j = x_j(1 - x_j) \sum_k \delta_k w_{jk}, \quad (3.9)$$

де k - номер шару, j - номер вузла.

- 5) Додати дельта ваги до кожного з

$$W_y(t + 1) = W_y(t) + \eta \delta_j x_i. \quad (3.10)$$

Всі кроки, окрім кроку 1, повторюються, поки помилка не зменшиться до бажаного значення. Після цього відрегульовані ваги зберігаються для довідки до алгоритму розпізнавання.

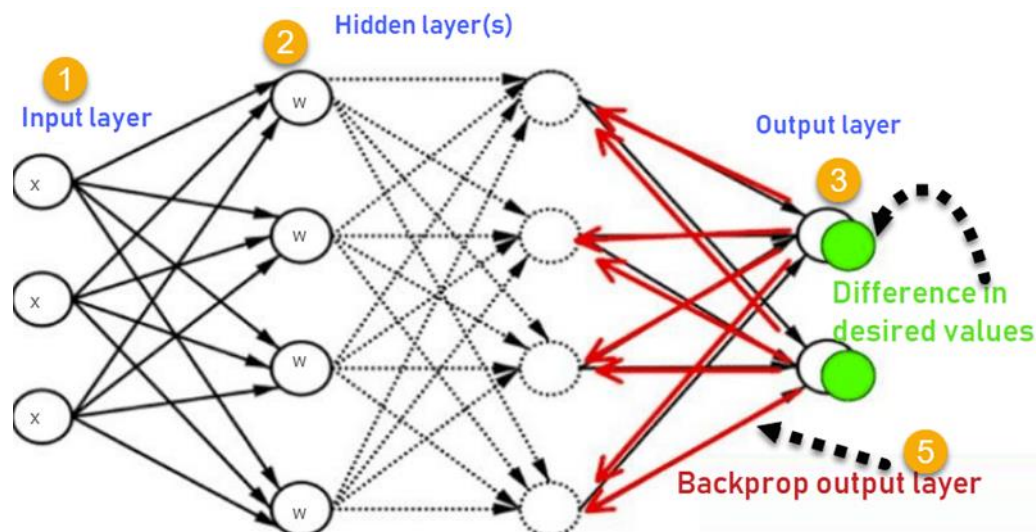


Рисунок 3.1 - Демонстрація алгоритму зворотного поширення помилки

3.3.1.2 Метод опорних векторів (Support Vector Machines)

SVM використовує алгоритми оптимізації, щоб знайти оптимальні межі між класами. Оптимальні межі повинні бути узагальненими, щоб не побачити вибірок з найменшими помилками серед усіх можливих меж, що розділяють класи, щоб мінімізуючи плутанину між класами.

1) Метод опорних векторів (SVM) — алгоритм машинного навчання, який базується на статистичній теорії. Було використано низку публікацій, що детально описують математичну постановку та розробку алгоритму SVM [7 - 9]. Індуктивним принципом, що стоїть за SVM, є структурна мінімізація ризиків (structural risk minimization, SRM). Ризик навчального алгоритму (R) обмежений сумою емпіричного ризику, оціненого з навчальних зразків (R_{emp}) та довірчого інтервалу (ψ): $R \leq R_{emp} + \psi$ [17]. Стратегією SRM є фіксація емпіричного ризику (R_{emp}) та мінімізація довірчого інтервалу (ψ), або максимізація відстані між роздільною лінією та найближчими точками даних. Роздільна гіперплощина відноситься до багатовимірного простору, який розділяє дані двох класів. Оптимальна роздільна гіперплощина максимізує межу між найближчими точками даних до площини. В даний час один класифікатор SVM здатний розділити лише два класи. Для розширення

цього методу необхідні інтеграційні стратегії для класифікації декількох класів;

2) Оптимальна роздільна лінія. Нехай треновані дані будуть розділені на два класи з k вибірками, які мають вигляд $(x_1, y_1), (x_k, y_k)$, де $x \in R^n$ - n -вимірний простір та $y \in \{+1, -1\}$ є мітками класів. Припустимо два класи можна розділити на дві гіперплощини, паралельні оптимальній лінії (рис. 2).

$$wx_i + b \geq 1 \quad (3.11),$$

$$wx_i + b \leq -1 \quad (3.12),$$

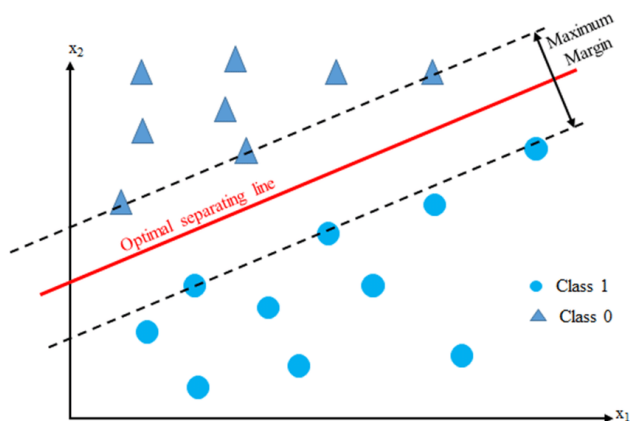


Рисунок 3.2 – Оптимальна роздільна лінія між відокремленими зразками даних

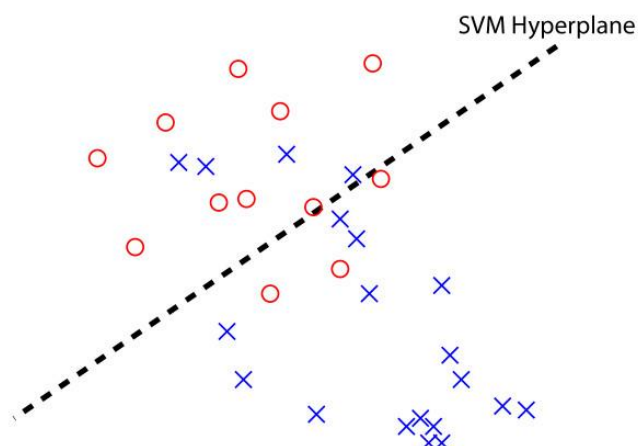


Рисунок 3.3 – Оптимальна роздільна лінія між нерозбірними зразками даних

де $w = (w_1 \dots w_n)$ – вектор з n елементів. Нерівності (3.11) і (3.12) можна об'єднати в єдину нерівність:

$$y_i[w'x_i + b] \geq 1. \quad (3.13)$$

Як показано на рисунках 3.2, 3.3, оптимальною розділювальною лінією є та, яка розділяє дані з максимальним запасом. Цю лінію можна знайти, мінімізуючи w , або виконавши наступну функцію:

$$F(w) = \frac{1}{2}(w'w) \quad (3.14)$$

Сідлові точки даного Lagrange дають рішення для вищезазначеної проблеми оптимізації:

$$L(w, b, \alpha) = \frac{1}{2}(w'w) - \sum_{i=1}^k \alpha_i y_i[w'x_i + b] - 1, \quad (3.15)$$

де $\alpha_i \geq 0$ - множники Лагранжа [20]. Вирішення цієї задачі оптимізації вимагає, щоб градієнт $L(w, b, \alpha)$ по відношенню до w і b зникав, даючи такі умови:

$$\sum_{i=1}^k y_i \alpha_i x_i = 0, \quad (3.16)$$

$$\sum_{i=1}^k \alpha_i y_i = 0. \quad (3.17)$$

Порівняльний аналіз різних алгоритмів машинного навчання

Проаналізуємо та порівняємо методи класифікації зображень, такі як штучні нейронні мережі, метод опорних векторів, нечіткі міри (fuzzy measures), генетичні алгоритми та їх комбінації щодо кількох параметрів.

Основна перевага нейронних мереж перед іншими методами полягає в тому, що вони можуть розпізнавати більш глибокі, іноді несподівані закономірності в даних. В процесі навчання нейрони здатні реагувати на отриману інформацію відповідно до принципів генералізації, тим самим вирішуючи поставлену перед ними задачу.

Метод опорних векторів є найбільш швидким методом знаходження вирішальних функцій. Метод чутливий до шумів і стандартизації даних.

Нечіткі міри мають перевагу виявлення стохастичних зв'язків для опису властивостей зображення. Однією з переваг нечітких мір є широкий спектр обчислювальних механізмів для реалізації синтезу кількох джерел інформації.

Генетичні алгоритми мають можливість знайти «достатньо швидко» «досить точний» розв'язок. Це робить генетичні алгоритми привабливими для використання при розв'язанні задач оптимізації.

Штучні нейронні мережі та метод опорних векторів дотримуються непараметричного підходу, тоді як нечіткі міри використовують стохастичні властивості для класифікації зображень. Вибір нелінійної межі є ефективним, коли дані мають лише декілька вхідних змінних в ANN і навпаки в SVM.

3.4 Опис методу розв'язання

3.4.1 Convolutional Neural Network

Згортова нейронна мережа (CNN) показала відмінні результати у багатьох задачах комп'ютерного зору та машинного навчання. На цю тему було опубліковано багато крутих робіт, і доступно досить багато високоякісних програмних пакетів із відкритим кодом CNN.

CNN корисна у багатьох програмах, особливо у виконанні завдань, пов'язаних із зображеннями [21]. До застосувань CNN належать класифікація зображень, семантична сегментація зображень, виявлення об'єктів на зображеннях тощо. Завдання дипломної роботи передбачає використання класифікації зображень.

Архітектура

CNN, як правило, приймає тензор третього порядку, наприклад, зображення з H рядків, W стовпців і 3 канали (R, G, B кольорові канали).

					ДП ІС-6212.0000-с.ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

Однак входи тензорів вищого порядку можуть CNN обробляти аналогічно. Потім вхідні дані послідовно проходять через серію обробки. Один крок обробки зазвичай називають шаром, який може бути шаром згортання, об'єднаним шаром, шаром нормалізації, повністю з'єднаним шаром, шаром втрат тощо.

Дамо абстрактний опис структури CNN:

$$x^1 \rightarrow w^1 \rightarrow x^2 \rightarrow \dots \rightarrow x^{L-1} \rightarrow w^{L-1} \rightarrow x^L \rightarrow w^L \rightarrow z, \quad (3.18)$$

Наведене вище рівняння 3.18 ілюструє, як CNN запускає шар за шаром при прямому поширенні. Вхід x^1 , як правило, зображення (3-тензорний порядок). Параметри, що беруть участь у обробці першого шару, ми позначаємо як тензор w^1 . Вихід першого шару - x^2 , який також виконує роль входу для обробки другого шару.

Ця обробка триває до тих пір, поки всі шари в CNN не будуть пройдені, виходом яких є x^L . Однак додається один додатковий шар для зворотнього поширення помилки - метод, який регулює ваги у CNN (рис. 3.1). Припустимо, що ця проблема є проблемою класифікації зображень C класів. Загальновживана стратегія полягає у вихідному шарі x^L у вигляді C розмірного вектора, i -й вузол позначає передбачення (зворотня ймовірність x^1 походить від i -го класу). Щоб зробити x^L функцією маси ймовірностей, ми можемо встановити обробку в $(L - 1)$ -му шарі як перетворення softmax x^{L-1} . В інших програмах вихідний шар x^L може мати інші форми та інтерпретації.

Останній шар - шар втрат. Припустимо, t - відповідне цільове значення для вхідного x^1 , тоді функцію витрат або збитків можна використовувати для вимірювання невідповідності між прогнозом CNN x^L та цільовим t . Наприклад, проста функція втрат може бути

$$z = \frac{1}{2} \|t - x^L\|^2, \quad (3.19)$$

					ДП ІС-6212.0000-с.ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

хоча зазвичай застосовуються більш складні функції втрат. Ця квадратна l_2 втрата може бути використана при проблемі регресії. У проблемі класифікації часто використовується перехресна ентропійна втрата. Основна істина в проблемі класифікації - категорична змінна t . Спочатку перетворюємо категоричну змінну t в C -вимірний вектор t . Тепер t , і x^L - функції масової ймовірності, а поперечна втрата ентропії вимірює відстань між ними. Отже, ми можемо мінімізувати перехресну ентропію. Рівняння (3.18) явно моделює функцію витрат як шар витрат, обробка якого моделюється як поле з параметрами w^L .

Треба зауважити, що деякі шари можуть не мати жодних параметрів, тобто w^i для деяких i може бути порожнім. Шар *softmax* є одним з таких прикладів.

3.4.2 Пряме поширення

Припустимо, всі параметри моделі CNN w^1, \dots, w^{L-1} були налаштовані, тоді ми готові використовувати цю модель для прогнозування. Передбачення результату передбачає лише запуск моделі CNN вперед, тобто в напрямку стрілок у рівнянні (3.18).

Візьмемо як приклад проблему класифікації зображень. Починаючи зі входу x^1 , ми передаємо його на обробку першого шару (поле з параметрами w^1) і отримаємо x^2 . У свою чергу, x^2 передається у другий шар і т. д. Наприкінці, ми досягаємо $x^L \in \mathbb{R}^C$, який оцінює остаточні ймовірності x^1 , що належать C категоріям. Ми можемо вивести прогноз CNN як

$$\operatorname{argmax}_i x_i^L. \quad (3.20)$$

Треба зауважити, що шар втрат не потрібен для прогнозування. Це корисно лише тоді, коли ми намагаємося відрегулювати параметри CNN, використовуючи набір навчальних прикладів. Тепер проблема полягає в тому, як дізнатися параметри моделі?

3.4.3 Стохастичний градієнтний спуск (SGD)

Як і у багатьох інших навчальних системах, параметри моделі CNN оптимізовані для мінімізації втрат z , тобто ми хочемо, щоб прогнозування моделі CNN відповідало міткам "основна істина".

Припустимо, для навчання таких параметрів подано один тренувальний приклад x^1 . Навчальний процес передбачає запуск мережі CNN в обох напрямках. Спочатку запускаємо мережу в прямому поширенні, щоб отримати x^L для досягнення прогнозування, використовуючи поточні параметри CNN. Замість виведення прогнозу нам потрібно порівняти прогноз із цільовою t , якій відповідає x^1 , тобто продовжувати виконувати перехід вперед до останнього рівня втрат. Наприкінці, ми досягаємо витрат z .

Витрата z – це показник, який визначає, як слід змінювати параметри моделі (оновлювати). І спосіб зміни параметрів SGD:

$$w^i \leftarrow w^i - \eta \frac{\partial z}{\partial w^i}. \quad (3.21)$$

Замітка про позначення. Ми використовуємо верхній індекс для позначення індексу шару. У рівнянні 15 знак \leftarrow неявно вказує, що параметри w^i (i -шару) оновлюються від часу t до $t + 1$.

У рівнянні (3.21), часткова похідна визначає швидкість зростання звідносно змін у різних вимірах w^i . Такий вектор часткових похідних називається градієнтом у математичній оптимізації. Отже, у невеликій локальній області навколо поточного значення w^i , переміщення w^i у напрямку, визначеному градієнтом, збільшить об'єктивне значення z . Щоб мінімізувати функцію втрат, нам слід оновити w^i вздовж протилежного напрямку градієнта. Це правило оновлення називається градієнтним спуском. Градієнтний спуск проілюстрований на рис. 3.4, на якому градієнт позначається, як g .

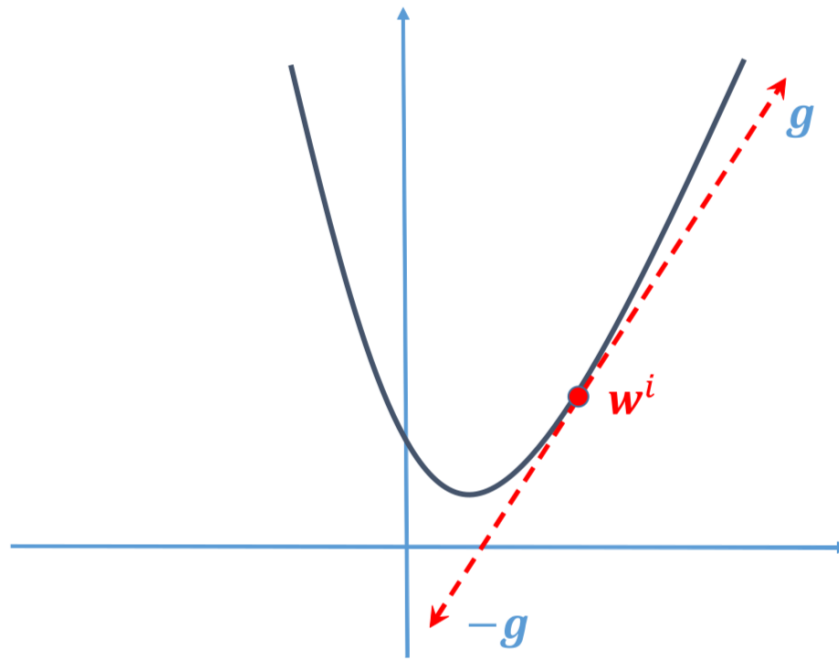


Рисунок 3.4 — Ілюстрація градієнтного спуску

Якщо рухатися занадто довго в негативному напрямку градієнта, тоді функція витрат може зрости. Отже, у кожному оновленні ми змінюємо параметри лише невеликою часткою від'ємного градієнта, керованого швидкістю навчання η . $\eta > 0$ зазвичай встановлюється невеликим значенням (наприклад, $\eta = 0,001$). Одне оновлення на основі x^1 зменшить втрати для цього конкретного прикладу навчання, якщо рівень навчання не надто великий. Однак цілком можливо, що це призведе до збільшення втрат деяких інших прикладів навчання. Отже, нам потрібно оновити параметри, використовуючи всі приклади навчання. Коли всі приклади набору даних будуть використані для оновлення параметрів, це означає, що одна епоха оброблена. Одна епоха, як правило, зменшить середні втрати на навчальному наборі, поки система навчання не зазнає переповнення. Отже, ми можемо повторювати епохи градієнтного спуску, поки функція витрат не почне збільшуватися, щоб отримати параметри CNN.

Градієнтний спуск може здатися простим у своїй математичній формі

					ДП ІС-6212.0000-с.ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

(3.21), але на практиці це дуже хитра операція. Наприклад, якщо ми оновлюємо параметри, використовуючи лише градієнт, обчислений лише з одного прикладу тренувань, ми спостерігатимемо нестабільну функцію витрат: середня втрата всіх прикладів тренувань буде підстрибувати вгору і вниз з дуже високою частотою. Це пояснюється тим, що градієнт оцінюється лише за допомогою одного прикладу тренування замість всього навчального набору. Оновлення параметрів з використанням градієнта, оціненого з невеликої підмножини прикладів тренувань, називають стохастичним градієнтним спуском. На відміну від SGD, заснованого на одному прикладі, ми можемо обчислити градієнт, використовуючи всі приклади тренувань, а потім оновити параметри. Однак ця пакетна обробка вимагає великої кількості обчислень, оскільки параметри оновлюються лише один раз в епоху, і, це недоцільно, коли кількість прикладів для навчання є великою.

Компромісом є використання міні-пакету тренувальної вибірки, обчислення градієнту за допомогою цієї міні-партії та оновлення параметрів. Наприклад, ми можемо встановити 32 або 64 приклади як міні-пакет. Стохастичний градієнтний спуск (SGD) є основним методом навчання параметрів CNN. При використанні міні-партії, вхід CNN стає тензором 4 порядку, наприклад, $H \times W \times 3 \times 32$, при розмірі міні-партії 32.

3.4.4 Шар входу, вихід та позначення та примітки

Тепер, коли архітектура CNN зрозуміла, перейдемо до типу шару ReLU. Припустимо, ми розглядаємо l -й шар, входи якого утворюють тензор x^l 3 порядку $x^l \in \mathbb{R}^{H^l \times W^l \times D^l}$. Таким чином, нам потрібен набір індексів-триплетів (i^l, j^l, d^l) , щоб знайти будь-який конкретний елемент у x^l . Триплет (i^l, j^l, d^l) відноситься до одного елемента в x^l , який знаходиться в d^l -му каналі, i в просторовому розташуванні (i^l, j^l, d^l) (в i^l -му ряду і j^l -му стовпчику). У навчанні CNN зазвичай використовується міні-пакетна стратегія. У такому випадку x^l стає тензором 4-ого порядку в $\mathbb{R}^{H^l \times W^l \times D^l \times N}$, де

					ДП ІС-6212.0000-с.ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

N - розмір міні-пакету. Для простоти вважатимемо, що $N = 1$.

У поясненні дотримуватимемось індексації з нуля, яка визначає, що $0 \leq i^l < H^l, 0 \leq j^l < W^l$ та $0 \leq d^l < D^l$.

На l -ому шарі функція трансформує вхід x^l в y , який є входом для наступного шару. Виходячи з цього, ута x^{l+1} по факту посилаються на один і той же об'єкт, і про це корисно не забувати. Підсумовуємо: розмір вихідного шару - $H^{l+1} \times W^{l+1} \times D^{l+1}$, елемент на виході проіндексований як триплет $(i^{l+1}, j^{l+1}, d^{l+1})$, $0 \leq i^{l+1} < H^{l+1}, 0 \leq j^{l+1} < W^{l+1}, 0 \leq d^{l+1} < D^{l+1}$.

3.4.5 Шар ReLU

Шар ReLU не змінює розмір входу, тому x^l та y мають однаковий розмір. Насправді Випрямлений Лінійний Вузол (Rectified Linear Unit, звідси назва ReLU) може розглядатися як усічення, що виконується індивідуально для кожного елемента на вході:

$$y_{i,j,d} = \max\{0, x_{i,j,d}^l\}, \quad (3.22)$$

де $0 \leq i \leq H^l \leq H^{l+1}, 0 \leq j < W^l = W^{l+1}$, та $0 \leq d < D^l = D^{l+1}$.

Всередині шару ReLU немає жодного параметра, отже, немає необхідності у вивченні параметрів цього шару.

Виходячи з рівняння, очевидно

$$\frac{dy_{i,j,d}}{dx_{i,j,d}^l} = \llbracket x_{i,j,d}^l > 0 \rrbracket, \quad (3.23)$$

де $\llbracket \cdot \rrbracket$ - показникова функція, яка дорівнює 1, якщо її аргумент є істинним та 0 — в іншому випадку.

Отже, маємо:

$$\left[\frac{\partial z}{\partial x^l} \right]_{i,j,d} = \begin{cases} \left[\frac{\partial z}{\partial x^l} \right]_{i,j,d} \\ 0 \end{cases} \quad (3.24)$$

					ДП ІС-6212.0000-с.ПЗ	Арк.
						52
Змн.	Арк.	№ докум.	Підпис	Дата		

Нагадаємо, що y є псевдонімом до x^{l+1} .

Строго кажучи, функція $\max(0, x)$ не диференційована при $x = 0$, отже рівняння (3.24) теоретично є трохи проблематичним. На практиці це не є проблемою, і ReLU безпечний у використанні.

Мета ReLU - посилити нелінійність CNN. Оскільки семантична інформація у зображенні (наприклад, людина та собака хаскі, що сидять поруч на лавці в саду), очевидно, є дуже нелінійним відображенням піксельних значень у вході, ми хочемо, щоб відображення від вводу CNN до його виходу також було б сильно нелінійним. Функція ReLU, хоч і проста, але є нелінійною функцією, як показано на рис. 3.5.

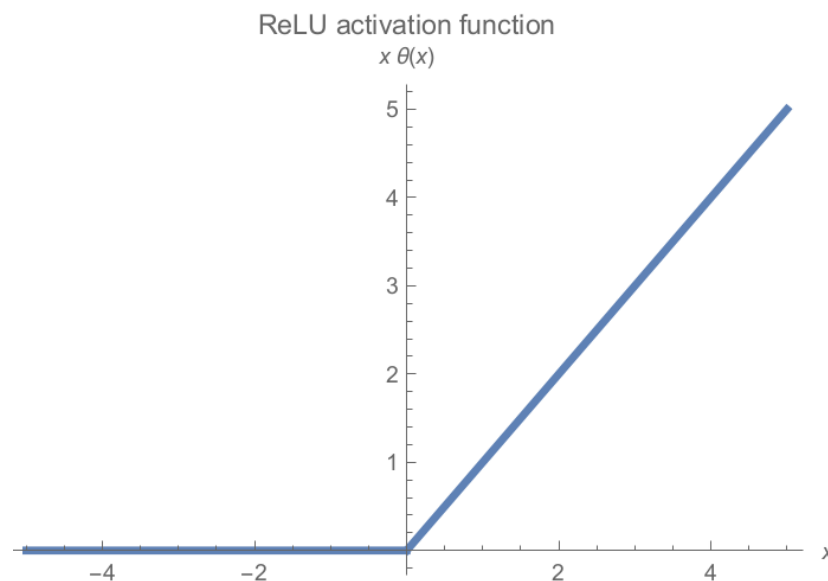


Рисунок 3.5 — Функція ReLU

Інші нелінійні перетворення використовуються в дослідженнях нейронної мережі для отримання нелінійності, наприклад, логістична сигмоїдна функція $y = \sigma(x) = \frac{1}{1+\exp(-x)}$. Однак логістична сигмоїда працює значно гірше, ніж ReLU у навчанні CNN. Зауважимо, що $0 < y < 1$, якщо використовується сигмоподібна функція, а $\frac{dy}{dx} = y(1 - y)$, маємо $\frac{dy}{dx} \leq \frac{1}{4}$. Отже,

у процесі зворотнього поширення помилки, градієнт $\frac{\partial z}{\partial z} = \frac{\partial z}{\partial y} \frac{dy}{dx}$ матиме набагато меншу величину, ніж $\frac{\partial z}{\partial y}$ (щонайбільше $\frac{1}{4}$). Іншими словами, сигмовидний шар призведе до значного зменшення величини градієнта, а після декількох сигмоїдних шарів градієнт зникне (тобто всі його компоненти будуть близькими до 0). Зникаючий градієнт ускладнює навчання на основі градієнта (наприклад, SGD).

З іншого боку, шар ReLU встановлює градієнт деяких функцій l -го шару на 0, але ці функції не активовані (тобто ми не зацікавлені в них). Для цих активованих функцій градієнт повертається назад без будь-яких змін, що вигідно для навчання SGD. Введення ReLU на заміну сигмоїду є важливою зміною CNN, що значно зменшує труднощі у навчанні параметрів CNN та підвищує його точність. Існують також більш складні варіанти ReLU, наприклад, параметрична ReLU і експоненціальна лінійна одиниця.

Висновок до розділу

У розділі математичного забезпечення було сформульовано змістовну та математичну постановку задачі, обґрунтовано причини вибору даного методу розв'язання та його переваги, а також описано цей метод

Зроблено порівняння штучних нейронних мереж та інших методів машинного навчання для класифікації зображень. Також, зроблено висновок, що нейромережевий підхід класифікації підвищує точність, є на сьогоднішній момент найкращим підходом для вирішення завдань такого типу, більш детальна інформація з окремих класів отримується за допомогою текстур. Розібрано архітектуру згорткової нейронної мережі, яка використовується для задач класифікації зображень. Визначено функції активації, наявні в моделі (для шару виходу використовуватиметься функція *softmax*, для інших шарів - ReLU).

					ДП ІС-6212.0000-с.ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

Розібрано алгоритми зворотнього поширення помилки (backpropagation, ВКР) та стохастичного градієнтного спуску (stochastic gradient descent, SGD), які використовуються при побудові моделей нейронних мереж.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Для реалізації програмного продукту було використано мову програмування Python та фреймворк TensorFlow. Були задіяні такі бібліотеки numpy, pandas, matplotlib, scikit-learn, opencv, tensorflow 2, keras, pillow мови Python.

Python (Пайтон або Пітон) — високорівнева мова програмування загального призначення, орієнтована на прискорення роботи розробника та має читабельний код. Синтаксис ядра мови мінімалістичний, в той же час стандартна бібліотека містить великий набір корисних функцій [22]. Python підтримує структурне, об'єктно-орієнтоване та функціональне програмування. Основні архітектурні особливості: динамічна типізація, автоматичне управління пам'яттю, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програми на модулі, які, в свою чергу, можуть об'єднуватися в пакети.

На даний момент Python є найпопулярнішою мовою програмування для досліджень та розробок у машинному навчанні. За даними Google Trends [23], інтерес до Python для машинного навчання піднявся на передові позиції у порівнянні з іншими мовами ML (рис. 4.1), такими як R, Java, Scala, Julia тощо, які значно відстають.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

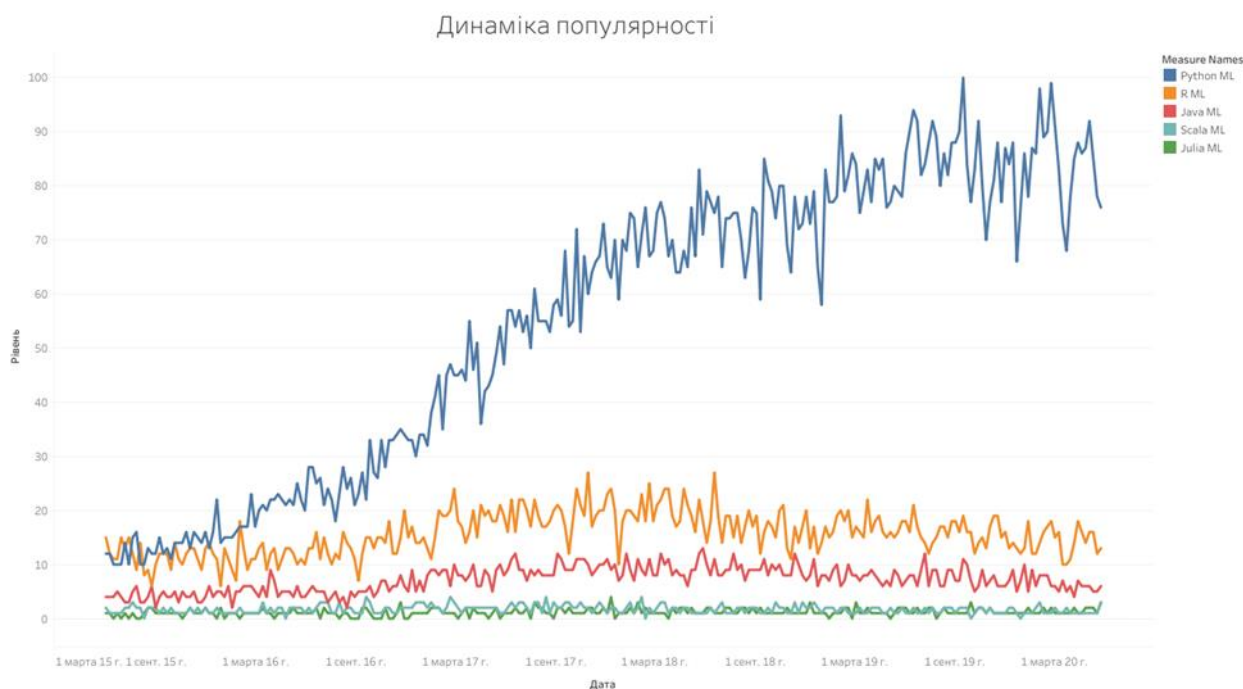


Рисунок 4.1 — Динаміка популярності використання мов програмування для машинного навчання

Причини, через які Python найкраще підходить для машинного навчання:

1. Python простий у використанні

Простота використання Python є однією з головних причин, чому він настільки популярний для машинного навчання. Синтаксис легко читається. Розробники можуть зосередитись на вирішенні проблеми, а не на технічних нюансах мови. На додаток до цього, Python є надзвичайно ефективним. Це дозволяє розробникам використовувати меншу кількість рядків коду. Код Python легко зрозумілий людям, що робить його ідеальним для реалізації моделей машинного навчання;

2. Python має велику кількість бібліотек та фреймворків

Python має сотні різних бібліотек та фреймворків, якими можуть користуватися розробники. Ці бібліотеки та фреймворки економлять час, що, в свою чергу, робить Python ще більш популярним.

Існує багато бібліотек Python, які використовуються для штучного інтелекту та машинного навчання [24]. Деякі з них наведені нижче:

- Keras - бібліотека з відкритим кодом, яка особливо зосереджена на експериментах з глибокими нейронними мережами;
- TensorFlow - безкоштовна бібліотека програмного забезпечення, яка використовується для багатьох програм машинного навчання, таких як нейронні мережі;
- Scikit-learn - безкоштовна бібліотека програмного забезпечення для машинного навчання, що пов'язане з різними алгоритмами класифікації, регресії та кластеризації. Також Scikit-learn можна використовувати разом із NumPy та SciPy;

3. Python має спільнотну та корпоративну підтримку

Python існує з 1990 року, і це є достатнім часом для створення підтримуючої спільноти. Завдяки цій підтримці учні, які навчаються на Python, можуть легко вдосконалити свої знання машинного навчання. Також корпоративна підтримка є дуже важливою складовою успіху Python для ML. Багато кращих компаній, таких як Google, Facebook, Instagram, Netflix, Quora тощо, використовують Python для своїх продуктів. Насправді Google несе відповідальність за створення багатьох бібліотек Python для машинного навчання, таких як Keras, TensorFlow тощо;

4. Python є портативним та розширюваним

Багато мовних операцій можна легко виконати на Python через його портативний та розширюваний характер. Є багато науковців, які вважають за краще використовувати графічний процесор (GPU) для навчання моделей ML на власних машинах, а портативний характер Python цілком підходить для цього.

Крім того, багато різних платформ підтримують Python, такі як

					ДП ІС-6212.0000-с.ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

Windows, Macintosh, Linux, Solaris і т.д.

4.2 Вимоги до технічного забезпечення

Даний програмний продукт представлений у вигляді віконного додатку, який запускається з консольного вікна. У склад комплексу повинен входити персональний комп'ютер користувача.

Мінімальні вимоги до характеристик компонентів технічного забезпечення, при яких значення часових параметрів Системи повинні відповідати пред'явленим в ТЗ вимогам:

- процесор з тактовою частотою не нижче 2 ГГц;
- остатній об'єм оперативної пам'яті (не менше 2 ГБ);
- жорсткий диск (не менше 40 ГБ);
- операційна система Ubuntu 18.04 і вище;
- Python 3.7;
- Tensorflow 2.1, matplotlib 3.1, numpy 1.18, opencv 3.4, scikit-learn 0.22;
- підключення до мережі Інтернет;
- набір даних додатків до атестатів та оцінок для класифікації.

Взаємодія із програмою відбувається за допомогою клавіатури та миші або тачпаду.

4.3 Архітектура програмного забезпечення

4.3.1 Діаграма класів

Структурна схема класів розміщена у графічному матеріалі.

Розроблений програмний продукт складається з чотирьох класів (рис. 4.2): Форма (Form), Дата Сет Зображень (ImageDataset), Інтерфейс Команд

					ДП ІС-62 12.0000-с.ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

(CommandAPI), Модель (Model).

Клас Form створює графічний інтерфейс користувача, який містить команди для виконання.

Клас CommandAPI є оболонкою, яка з одного боку приймає команди для виконання з інтерфейсу, з іншого - звертається до методів з класів Model та ImageDataset.

Клас ImageDataset завантажує дані з директорії, виконує попередню обробку зображень, зменшує розмір, перетворює в чорно-біле для побудови моделі, розділяє дані на тренувальні та тестові, записує результат класифікації у csv-файл.

Клас Model будує та тренує згорткову нейронну мережу, зберігає історію регуляції вагів моделі, класифікує зображення та виводить на екран графік точності побудови моделі по епохам.

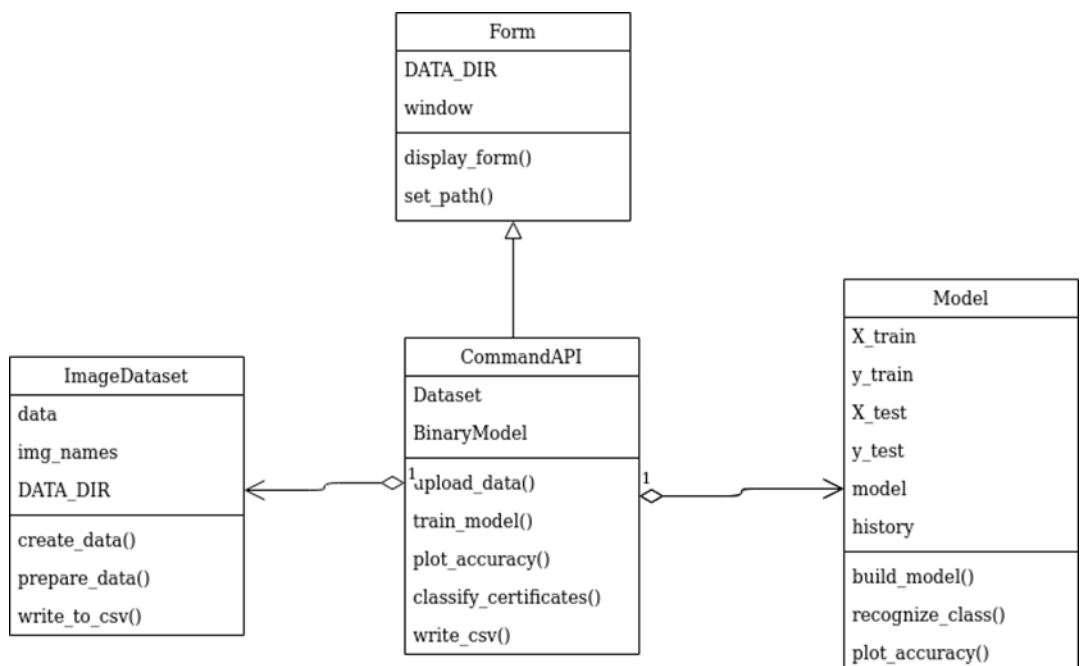


Рисунок 4.2 — Діаграма класів

4.3.2 Діаграма послідовностей

Структурна схема послідовності розміщена у графічному матеріалі.

Діаграма складається з чотирьох елементів: “:Form”, “:CommandAPI”, “:ImageDataset”, “:Model”.

Пройдемо по сценарію, який передбачений для виконання цілей поставлених перед користувачем.

Спочатку користувач запускає скрипт у консольному вікні для запуску застосунку, після чого об’єкт класу Form створює форму - графічний інтерфейс користувача. Після цього юзер повинен увести шлях до директорії із зображеннями. При помилці програма повідомить про це та запропонує ввести шлях заново. Далі користувач натискає на кнопку “Upload data” для підгрузки даних. Якщо операцію було проведено успішно можна переходити до створення класифікатору.

Для створення класифікатору треба створити та натренувати модель на даних, які були завантажені. Щоб це виконати необхідно натиснути на кнопку “Train the model”. Після завершення навчання модель буде збережено.

Наступна дія не є обов’язковою, вона лише дає можливість юзеру візуально оцінити точність по етапам побудови моделі. Для цього треба натиснути кнопку “Display model accuracy”.

Далі за сценарієм користувач натискає “Classify certificates on correct/incorrect”, застосунок розподіляє додатки за класами, за допомогою попередньо побудованої моделі.

Для збереження результату класифікації особа, що взаємодіє із системою повинна натиснути кнопку з написом “Write results in csv-file”.

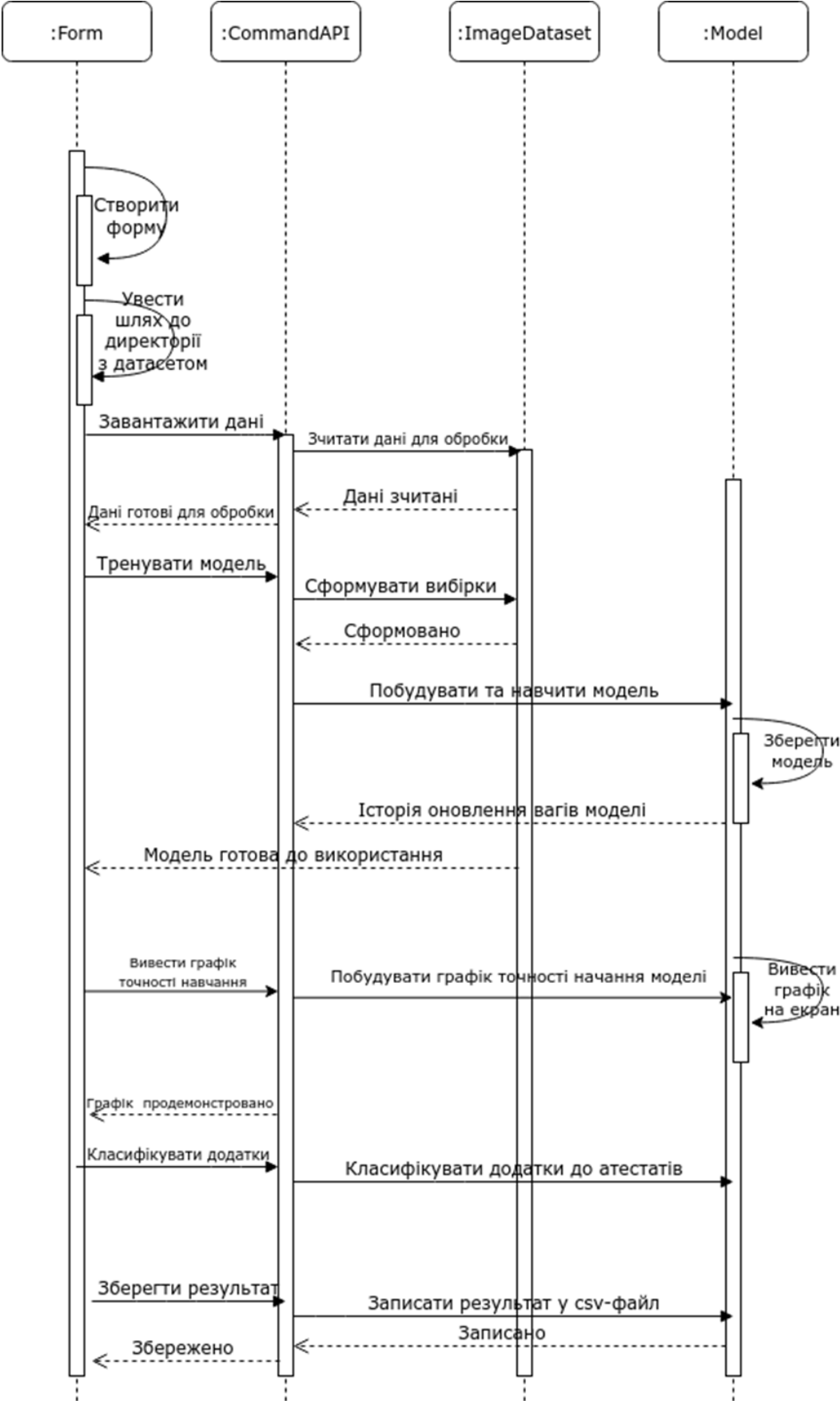


Рисунок 4.3 – Діаграма послідовностей

4.3.3 Діаграма компонентів

Система містить такі компоненти: Форма (Form), Інтерфейс Команд (CommandAPI), Модель (Model), Дата Сет Зображень (ImageDataset).

Компонент Форма (Form) представляю собою інтерфейс, за допомогою якого користувач здатний виконати необхідні дії.

Компонент Інтерфейс Команд (CommandAPI) слугує оболонкою, яка містить всі функції програми в собі, що є зручним для додавання в графічний інтерфейс.

Компонент Модель (Model) виконує команди подані з Інтерфейсу Команд (CommandAPI), використовуючи компонент Дата Сет зображень (ImageDataset).

Компонент Дата Сет Зображень (Image Dataset) надає компоненту Модель (Model) необхідний набір зображень для навчання.

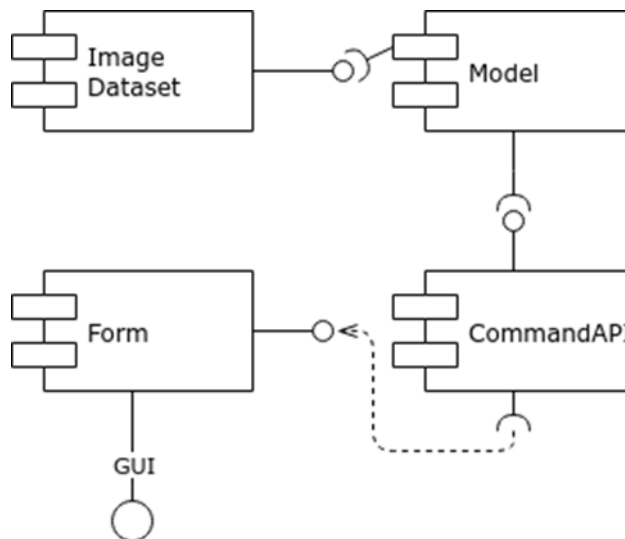


Рисунок 4.4 – Діаграма компонентів

4.3.4 Специфікація функцій

Таблиця 4.1 – Специфікація функцій програми

Назва функції	Опис
Клас ImageDataset	
create_data()	Зчитати зображення двох класів (correct/incorrect) у цифровий вигляд із файлу, конвертувавши у зображення у двухкольорове чорно-біле та змінивши зображення до оптимального для навчання розміру.
prepare_data()	Розділити дані на features/labels, нормалізувати їх. Features та labels розділити на тренувану та тестову вибірки.
write_to_csv(pred)	Записати результат у csv-файл у форматі [«ім'я додатку», «клас», «номер класу»].
Клас Model	
build_model()	Побудувати, скомпілювати та натренувати згорткову нейронну мережу. Зберегти ваги моделі.
recognize_class()	Класифікувати набір зображень додатків до атестатів.
plot_accuracy()	Побудувати графік точності моделі по епохам тренування.

Продовження таблиці 4.1

Клас CommandAPI	
upload_data (DATA_DIR)	Створити об'єкт класу ImageDataset та зчитати датасет зображень з файлу та підготувати для подальшої обробки.
train_model()	Розділити дані на features/labels, нормалізувати їх. Features та labels розділити на тренувану та тестову вибірки. Створити об'єкт класу Model, натренувати згорткову нейронну мережу та зберегти ваги моделі для класифікації.
plot_accuracy()	Побудувати графік точності моделі по епохам тренування.
classify_certificates()	Класифікувати набір зображень додатків до атестатів.
write_csv()	Записати результат у csv-файл у форматі [«ім'я додатку», «клас (передбачення)», «номер класу»].

Продовження таблиці 4.1

Клас Form	
display_form()	<p>Відобразити форму з командами для виконання:</p> <ol style="list-style-type: none"> 1. Тренувати модель; 2. Продемонструвати точність моделі; 3. Класифікувати додатки до атестатів на відповідні / невідповідні; 4. Записати результат у csv-файл.
set_path()	Отримати шлях до директорії з датасетом із поля вводу у формі.

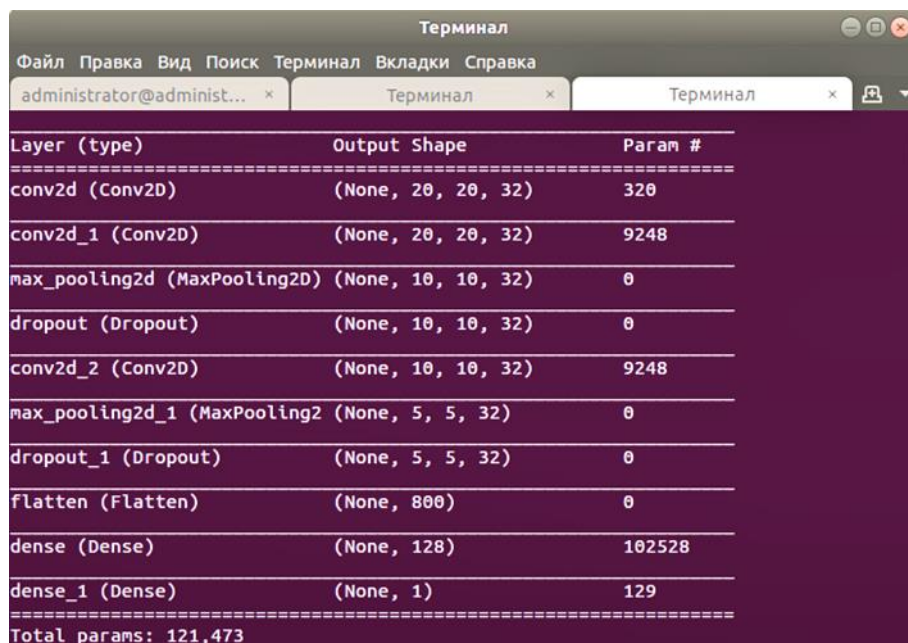
4.4 Опис звітів

Під час тестування моделей згорткової нейронної мережі, на консоль виводяться її архітектура CNN (рис. 4.5), кількість шарів та вузлів, а також зміни точності моделі під час навчання для тренованих та тестових даних на кожній епосі (рис. 4.6). Це дає можливість оцінити ефективність побудованої моделі.

На рис. 4.6 можемо побачити, що навчання бінарного класифікатора складається з 60 епох, фінальне значення функції витрат для тренованих даних - 0,0457, точності моделі - 98,38%; для тестових даних: функція витрат - 0,2865, точність моделі - 94,78%.

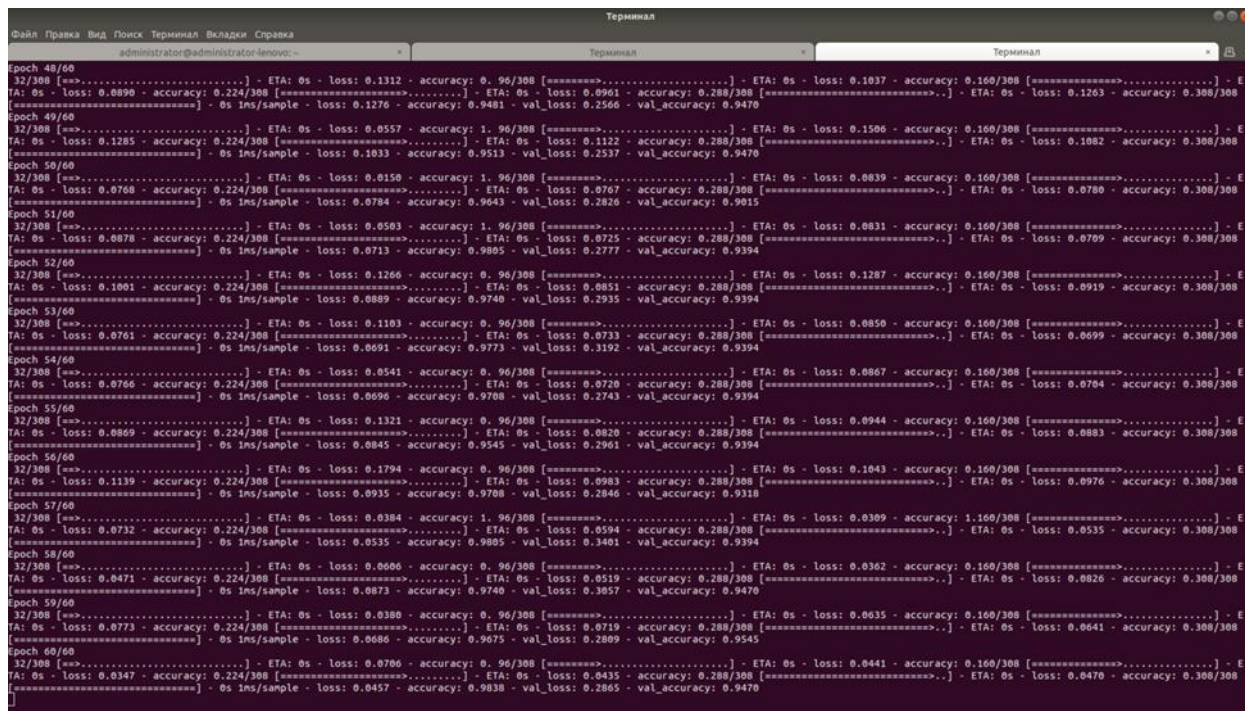
Навчальна вибірка для класифікації рукописних зображень оцінок складається з 477 зображень (рис. 4.2). Найпопулярніша оцінка — десять. Оцінок “один” та “два” в додатках виявлено не було. Кількість класів — 10.

Після класифікації результат записується у csv-файл. Файл містить такі поля: id (унікальний ідентифікатор), name (унікальне ім'я додатку до атестата), category (передбачений клас), category_num (індекс передбаченого класу), correct_category_num (дійсний індекс класу).



Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 20, 20, 32)	320
conv2d_1 (Conv2D)	(None, 20, 20, 32)	9248
max_pooling2d (MaxPooling2D)	(None, 10, 10, 32)	0
dropout (Dropout)	(None, 10, 10, 32)	0
conv2d_2 (Conv2D)	(None, 10, 10, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 32)	0
dropout_1 (Dropout)	(None, 5, 5, 32)	0
flatten (Flatten)	(None, 800)	0
dense (Dense)	(None, 128)	102528
dense_1 (Dense)	(None, 1)	129
Total params: 121,473		

Рисунок 4.5 — Архітектура шарів нейронної мережі



Epoch	loss	accuracy	val_loss	val_accuracy
48/60	0.1312	0.96/308	0.1037	0.160/308
49/60	0.0890	0.224/308	0.0961	0.288/308
50/60	0.1285	0.224/308	0.1122	0.288/308
51/60	0.0557	1.96/308	0.1506	0.160/308
52/60	0.0150	1.96/308	0.0839	0.160/308
53/60	0.0768	0.224/308	0.0767	0.288/308
54/60	0.0583	1.96/308	0.0831	0.160/308
55/60	0.0078	0.224/308	0.0725	0.288/308
56/60	0.1266	0.96/308	0.1287	0.160/308
57/60	0.1001	0.224/308	0.0851	0.288/308
58/60	0.0761	0.224/308	0.0733	0.288/308
59/60	0.0541	0.96/308	0.0867	0.160/308
60/60	0.0766	0.224/308	0.0720	0.288/308
61/60	0.0869	0.224/308	0.0820	0.288/308
62/60	0.1321	0.96/308	0.0944	0.160/308
63/60	0.0794	0.224/308	0.0719	0.288/308
64/60	0.0606	0.96/308	0.0362	0.160/308
65/60	0.0471	0.224/308	0.0519	0.288/308
66/60	0.0773	0.224/308	0.0635	0.160/308
67/60	0.0347	0.224/308	0.0435	0.288/308
68/60	0.0457	0.9838	0.2865	0.9470

Рисунок 4.6 — Точність навчання моделі

	A	B	C	D	E
1	id	name	category	category_num	correct category_num
2	1	Atestat_3711285.jpg	correct	0	0
3	2	Atestat_3708986.jpg	correct	0	0
4	3	Atestat_4060475.jpg	correct	0	0
5	4	Atestat_3711573.jpg	correct	0	0
6	5	Atestat_3711688.jpg	incorrect	1	0
7	6	Atestat_4043203.jpg	incorrect	1	1
8	7	Atestat_3708646.jpg	correct	0	0
9	8	Atestat_4258280.jpg	correct	0	0
10	9	Atestat_3711223.jpg	correct	0	0
11	10	Atestat_3711336.jpg	correct	0	0
12	11	Atestat_3711180.jpg	correct	0	0
13	12	Atestat_4258586.jpg	correct	0	0
14	13	Atestat_3712914.jpg	correct	0	0
15	14	Atestat_4177395.jpg	incorrect	1	1
16	15	Atestat_4272694.jpg	incorrect	1	1
17	16	Atestat_3712195.jpg	correct	0	0
18	17	Atestat_3707599.jpg	correct	0	0
19	18	Atestat_3711816.jpg	correct	0	0
20	19	Atestat_3707814.jpg	correct	0	0
21	20	Atestat_3707443.jpg	correct	0	0
22	21	Atestat_3984851.jpg	incorrect	1	1
23	22	Atestat_4259510.jpg	correct	0	0
24	23	Atestat_4060219.jpg	correct	0	0
25	24	Atestat_3708221.jpg	correct	0	0
26	25	Atestat_3708527.jpg	correct	0	0
27	26	Atestat_4258436.jpg	correct	0	0
28	27	Atestat_3935663.jpg	incorrect	1	1
29	28	Atestat_3930594.jpg	correct	0	1
30	29	Atestat_3710070.jpg	correct	0	0
31	30	Atestat_3710077.jpg	correct	0	0
32	31	Atestat_4258500.jpg	correct	0	0

Рисунок 4.8 — Приклад csv-файлу з результатом розпізнавання

Висновок до розділу

Програмний продукт реалізовано мовою Python. На даний момент Python є найпопулярнішою мовою програмування для досліджень та розробок у машинному навчанні. Мова проста у використанні, має велику кількість бібліотек та фреймворків, має спільноту та корпоративну підтримку, є портативною та розширюваною. Програмний продукт представлений у вигляді віконного додатку, який запускається з консольного вікна. У розділі детально описані складові частини і елементи діаграм, схеми яких розміщені у графічному матеріалі. Було побудовано такі діаграми: класів, послідовності, компонентів. Описано специфікації функцій класів.

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Розглянемо функції користувача, які було реалізовано у програмному продукті. Користувач має такі функції в рамках системи:

- завантажити дата сет зображень додатків;
- конвертувати зображення в чорно-білий формат з кольорового або відтінків сірого (порогова бінаризація);
- побудувати та натренувати модель на персональних даних;
- побудувати та продемонструвати графік точності навчання для кожної епохи для тренувальної та тестувальної вибірок;
- зберегти натреновану модель (відрегульовані ваги) у файл формату .h5;
- завантажити готову модель або ту, яку було збережено;
- класифікувати зображення на відповідність для подальшої обробки (додаток до атестату чи стороннє зображення);
- класифікувати зображення рукописних оцінок (HWR, Handwriting Recognition);
- записати результат класифікації у csv-файл у форматі [«ім'я додатку» / «унікальний номер зображення оцінки», «клас (передбачення)», «номер класу»].

Для запуску програмного продукту повинен бути встановлений Python версії 3.7. Встановити реліз можна з офіційного сайту [24]:

Інструкцію із алгоритмом дій для встановлення релізу можна знайти на сайті [14]. Під час інсталяції необхідно поставити прапорець поряд з пунктом “add Pyton 3.7 to PATH”, для того щоб додати директорію з файлами до змінної оточення.

					ДП ІС-6212.0000-с.ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 5.1 - Шлях змінної оточення

Для початку роботи з програмним продуктом користувач повинен запустити скрипт мови Python після чого відкриється форма з кнопками, де користувач повинен обрати команду. Виконання програми супроводжується використанням консольного вікна, у якому користувач бачить поточний статус виконання команд.

Перед запуском скрипту користувач повинен впевнитись у тому, що встановлено усі необхідні бібліотеки мови Python для коректної роботи програми: tensorflow 2.1, matplotlib 3.1, numpy 1.18, opencv 3.4, scikit-learn 0.22.

Для того щоб встановити бібліотеку необхідно увести команду [14]:

```
pip install lib_name x.y,
```

де *lib_name* - назва бібліотеки, *x.y* - версія бібліотеки.

Для запуску програми треба запустити скрипт у консольному вікні:

```
python file_name.py,
```

де *file_name* - назва файлу для запуску.

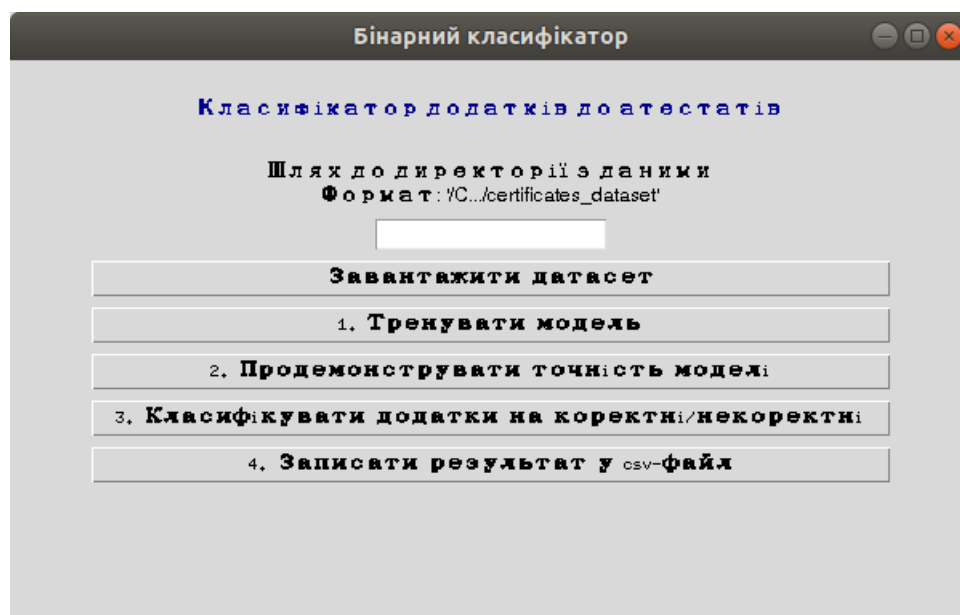


Рисунок 5.2 - Інтерфейс додатку

Для завантаження даних необхідно вказати шлях до директорії із зображеннями додатків. Директорія повинна складатися з двох папок: “correct” та “incorrect” (рис. 5.3) - навчальної та тренувальної вибірок.

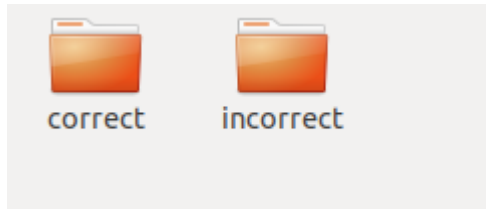


Рисунок 5.3 - Директорія із зображеннями додатків

Команда “1. Тренувати модель” розділяє дані на features/labels, нормалізує їх. Features та labels розділяються на треновану та тестову вибірки. Створює об’єкт класу Model, тренує згорткову нейронну мережу (рис. 4.5) та зберігає ваги моделі для класифікації.

Команда “2. Продемонструвати точність моделі” будує графік точності моделі для тренувальних та тестувальних даних для бінарної класифікації по епохам тренування (рис. 5.4).

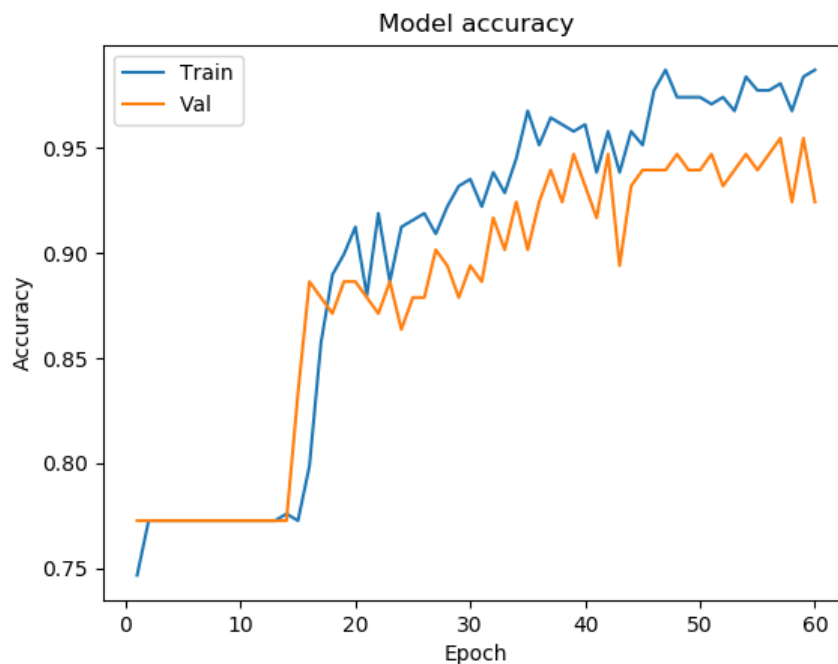


Рисунок 5.4 - Графік точності моделі бінарної класифікації

Команда “3. Класифікувати додатки на коректні / некоректні” класифікує набір зображень додатків до атестатів. А команда “4. Записати результат у csv-файл” записує результат класифікації у csv-файл у форматі [«ім’я додатку» / «унікальний номер зображення оцінки», «клас (передбачення)», «номер класу»] (рис. 4.8).

Для тренування багатокритеріальної моделі створюється набір даних рукописних оцінок з додатків (від трьох до дванадцяти). Оцінок «один», «два» не було знайдено. Приклади підготовлених даних зображено на рис. 5.5, 5.6. Дані створено власноруч. Якщо користувач бажає побудувати модель на власних даних, він повинен у кореневій папці програми створити папку, в якому буде знаходитися його власний набір даних. Після цього у цій папці треба створити ще один рівень папок, які будуть позначати класи зображень (назва папки – назва класу). Назви цих класів треба додати до змінної «CATEGORIES» у файлі «mark.py». Після цього можна запуснути скрипт та натренувати модель для власних потреб.



Рисунок 5.5 – Зображення оцінки «десять»

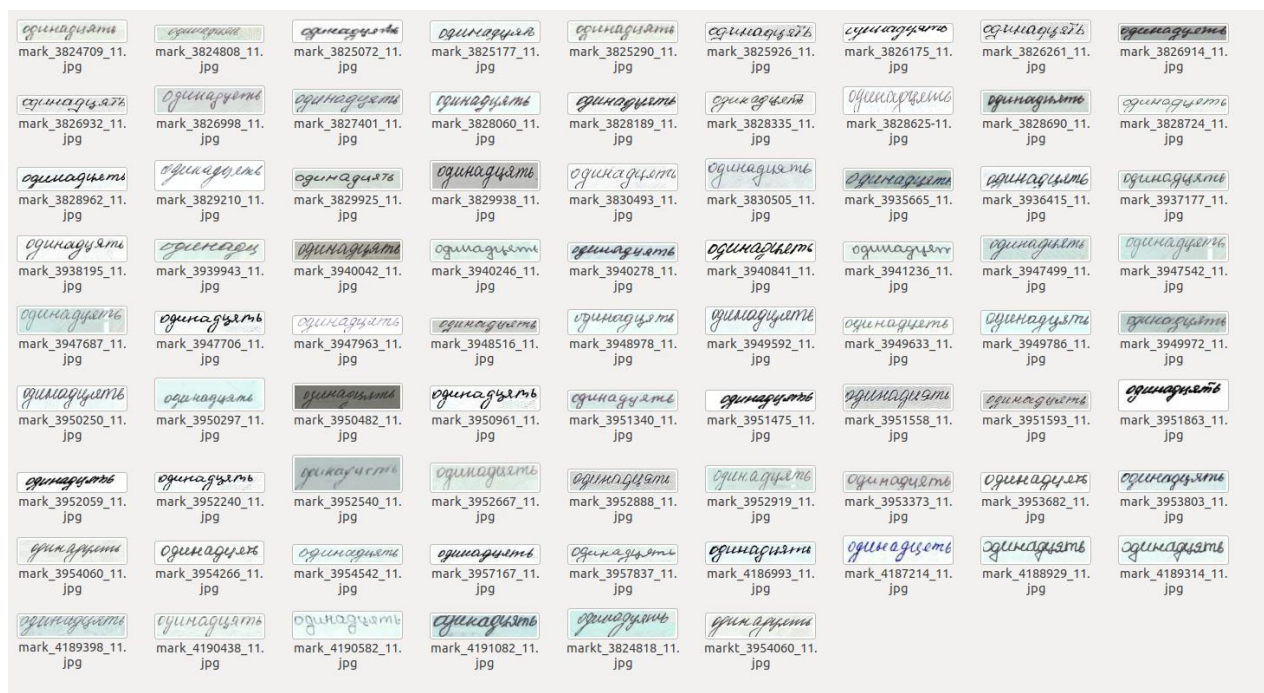


Рисунок 5.6 – Зображення оцінки «одиннадцать»

Графік точності моделі для тренувальних та тестувальних даних для класифікації оцінок, написаних рукописом по епохам продемонстровано на

					ДП ІС-6212.0000-с.ПЗ	Арк.
						73
ЗМН.	Арк.	№ докум.	Підпис	Дата		

рис. 5.7.

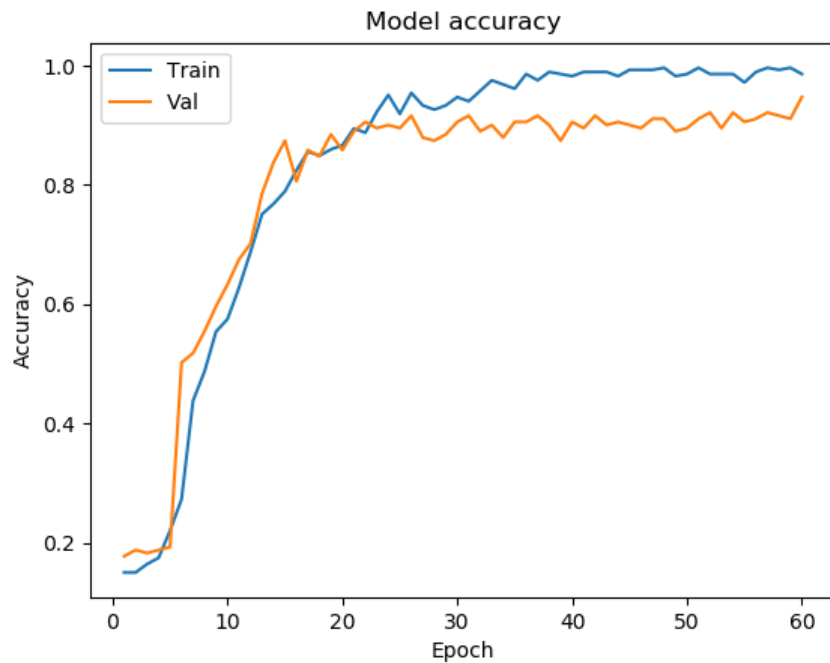


Рисунок 5.7 - Графік точності моделі багатокритеріального розпізнавання

Детальну інформацію по достовірності класифікації зображень можна побачити на матриці помилок (confusion matrix) на рис. 5.8.

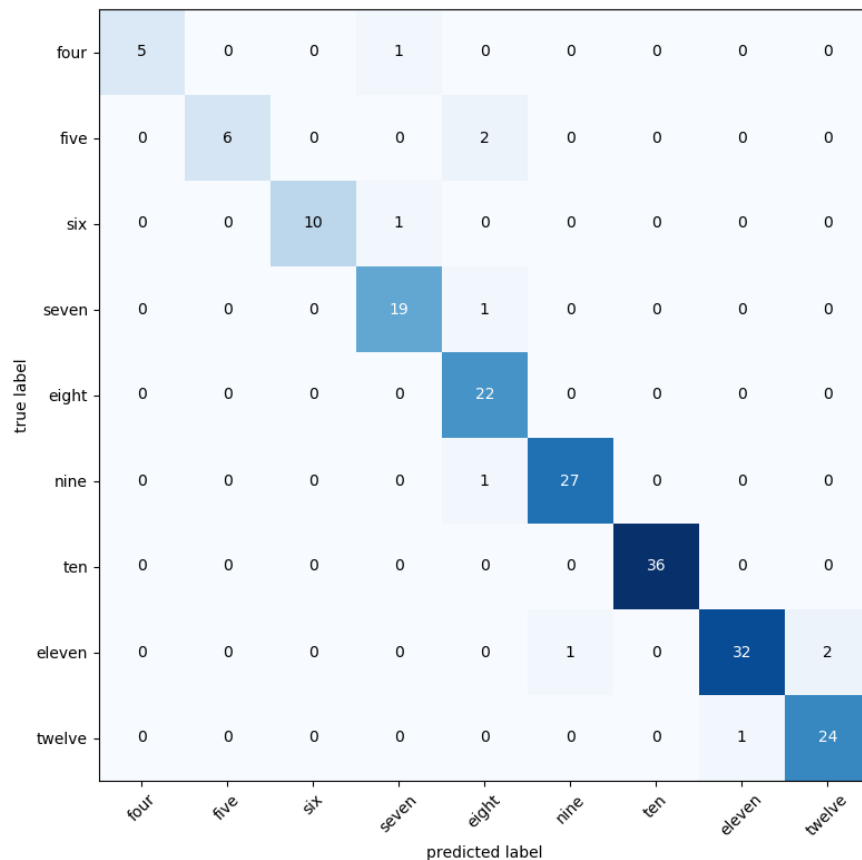


Рисунок 5.8 - Матриця помилок класифікації

Після класифікації результат записується у csv-файл (за замовчуванням – «mark_classification.csv»). Приклад формату, у якому подано результат зображено на рис. 5.9. У першому стовпці – унікальний порядковий ідентифікатор, у другому – назва зображення, у третьому – прогнозована оцінка, і у останньому – бажаний (правильний) результат.

id	name	pred_mark	mark
1	mark_3956848_5.png	8	5
2	mark_3952667_8.jpg	8	8
3	mark_3935665_11.jpg	11	11
4	mark_3940156_6.jpg	6	6
5	mark_3951340_11.jpg	11	11
6	mark_3957933_4.jpg	4	4
7	mark_3828625_8.jpg	6	8
8	mark_3706504_10.jpg	10	10
9	mark_3735827_10.jpg	10	10
10	mark_3829210_6.jpg	6	6
11	mark_3952540_8.jpg	7	8
12	mark_3826932_11.jpg	11	11
13	mark_3825072_12.jpg	11	12
14	mark_3954116_7.jpg	7	7
15	mark_3939943_10.jpg	10	10
16	mark_3956700_5.jpg	5	5
17	mark_3948119_8.jpg	8	8
18	mark_3941236_12.jpg	11	12
19	mark_4188929_11.jpg	11	11
20	mark_3947577_9.jpg	9	9
21	mark_4188929_7.jpg	7	7
22	mark_3829938_7.jpg	7	7
23	mark_3949592_8.jpg	8	8
24	mark_3953803_10.jpg	10	10
25	mark_3949972_12.jpg	12	12
26	mark_3940836_10.jpg	10	10

Рисунок 5.9 – Результат класифікації зображень рукописних оцінок

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

У процесі випробувань необхідно протестувати такий функціонал:

- працездатність побудови моделей за допомогою нейронних мереж для класифікації зображень;
- визначити при якому розмірі зображень, при зниженні якості для навчання, модель працює найкраще;
- визначити оптимальний розподіл тренувальних та тестових даних для валідації при навчанні моделі;
- визначити оптимальну кількість епох для навчання моделі;

					ДП ІС-6212.0000-с.ПЗ	Арк.
						76
Змн.	Арк.	№ докум.	Підпис	Дата		

- функціональна працездатність графічного інтерфейсу користувача;
- зручність роботи із графічним інтерфейсом користувача;
- збереження результатів виконання програми;
- перевірка відповідності функцій програми вимогам технічного завдання.

5.2.2 Загальні положення

- ГОСТ 34.603-92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

Опишемо контрольні приклади тестування різних складових готового продукту.

Таблиця 5.1 - Завантаження даних

Мета тесту	Перевірка коректності завантаження даних набору зображень
Початковий стан	Інтерфейс користувача
Вхідні дані	Некоректна адреса набору зображень
Схема проведення тесту	У поле "Path to image dataset:" увести некоректну адресу директорії, просто будь-які символи або залишити порожнім

Продовження таблиці 5.1

Очікуваний результат	Застосунок повідомить про некоректно введену адресу при спробі зчитати датасет та запропонує увести значення знову
-------------------------	---

Таблиця 5.2 - Побудова моделі

Мета тесту	Перевірка коректності побудови, тренування та збереження моделі згорткової нейронної мережі
Початковий стан	Інтерфейс користувача
Вхідні дані	Датасет зображень додатків
Схема проведення тесту	Натиснути кнопку “Train the model”. Зберегти результат у файл формату .h5 та зберегти результат розпізнавання у csv-файл.
Очікуваний результат	Відображення процесу навчання моделі у консольному вікні, збережена модель та csv-файл у директорії

Таблиця 5.3 - Параметри моделі

Мета тесту	Визначення оптимальних параметрів моделі нейронної мережі
Початковий стан	Інтерфейс користувача
Вхідні дані	Датасет зображень додатків, розмір стиснення зображення, кількість епох навчання, відсоток валідаційних тестових даних.
Схема проведення тесту	Визначити при якому розмірі зображень, при зниженні якості для навчання, модель працює найкраще; визначити оптимальний розподіл тренувальних та тестових даних для валідації при навчанні моделі; визначити оптимальну кількість епох для навчання моделі.
Очікуваний результат	Для порівняння результатів будемо користуватися графіками точності навчання.

Усі випробування пройдено успішно.

Висновок до розділу

У розділі були розглянуті функції користувача, які було реалізовано у програмному продукті. Було роз'яснено, які інструменти повинні бути встановлені на ПК для запуску застосунку та інструкції для їх встановлення. Описані мета випробувань, загальні положення та результати. Були

перевірені коректність завантаження датасету в систему, перевірка коректності побудови, тренування та збереження моделі згорткової нейронної мережі та визначено оптимальні параметри моделі нейронної мережі.

					ДП ІС-62 12.0000-с.ПЗ	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗАГАЛЬНІ ВИСНОВКИ

Дипломна бакалаврська робота розглядає проблеми, які виникають при обробці зображень додатків до атестатів абітурієнтів під час вступної кампанії, на прикладі аналізу більше 10 тисяч зображень додатків до атестата, поданих в КПІ ім. Ігоря Сікорського в 2018 році. Наведені типові приклади зображень документів і зроблені висновки про можливість автоматичного визначення середнього балу на підставі сканованих копій.

У роботі представлені сучасні методи і інструменти розпізнавання та класифікації рукописного тексту на зображенні. Продемонстровано класифікацію зображень за змістом. Реалізовано на мові Python, за допомогою фреймворку TensorFlow, згорткові нейронні мережу, які класифікують зображення оцінок та визначають «проблемні» зображення.

					ДП ІС-62 12.0000-с.ПЗ	Арк.
						81
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ

1. Методи офлайн-розпізнавання рукописного тексту на прикладі задачі автоматичного визначення середнього балу додатку до атестата абітурієнта / І. Д. Марченков // Всеукраїнська науково-практична конференція молодих вчених та студентів «Інформаційні системи та технології управління – ІСТУ-2020». Секція кафедри автоматизованих систем обробки інформації і управління. Матеріали конференції. – Київ. – 2020. 24, 30 квітня 2020р. – с. 105-110.
2. Умови прийому на навчання до вищих навчальних закладів України у 2018 році. URL: <https://zakon5.rada.gov.ua/laws/show/z1397-17/page#n17>. (дата звернення: 12.02.2019).
3. Статистика ВНЗ по количеству поданных заявлений. URL: <https://grade.ua/uk/news/interesnaya-statistika-vstupitelnoj-kampanii-2018/>. (дата звернення : 05.04.2019).
4. Особенности автоматического определения среднего балла аттестата абитуриента при обработке изображений аттестатов / Т. С. Ладогубец, П. Л. Литвиненко, Р. И. Сегол, А. Д. Финогенов // [Сучасні проблеми моделювання](#). - 2019. - Вип. 15. - С. 118-127. - Режим доступу: http://nbuv.gov.ua/UJRN/cpm_2019_15_18.
5. Ten awesome things you can do with Google Lens [Електронний ресурс] : [Веб-сайт]. Режим доступу: <https://www.androidguys.com/tips-tools/ten-fun-things-google-lens/>, 2018.
6. Aipoly vision is a truly visionary app [Електронний ресурс]: [Веб-сайт]. Режим доступу: https://mediaaccess.org.au/latest_news/blind-or-vision-impaired/aipoly-vision-is-a-truly-visionary-app#:~:text=%27Aipoly%20Vision%27%20is%20a%20very,each%20object%20to%20the%20user, 2017.

7. Leafsnap: An Electronic Field Guide [Электронный ресурс]: [Веб-сайт]. Режим доступа: <http://leafsnap.com>, 2020.
8. Pen to Print - Handwriting OCR [Электронный ресурс]: [Веб-сайт]. Режим доступа: <https://apps.apple.com/us/app/pen-to-print-handwriting-ocr/id1308003011>, 2020.
9. Технические требования ABBYY FineReader 14 [Электронный ресурс]. Режим доступа: <https://web.archive.org/web/20170811183907/https://www.abbyy.com/ru-ru/finereader/tech-specs/>, 2017.
10. Integrate AI-powered OCR features into your applications [Электронный ресурс]: [Веб-сайт]. Режим доступа: https://www.abbyy.com/en-ee/ocr-sdk/?utm_source=ocr4linux301&utm_medium=redirect&utm_campaign=ocr4linux301. 2020.
11. IT opportunities & challenges in Russia / Vladimir Radyuhin. – 2016. Режим доступа: <https://www.thehindu.com/todays-paper/tp-opinion/IT-opportunities-and-challenges-in-Russia/article15146388.ece>.
12. Теория распознавания образов / Вапник В. Н., Червоненкис А. Я. – М.: Наука. 1974.
13. Нейронные сети: полный курс, 2-е издание. / Хайкин С. – М.: Вильямс. 2008
14. Обзор теории интеллектуального анализа данных на базе нейронных сетей / Ерофеева В. А. - СПб.:СПбГУ, 2015. с. 11-12.
15. Artificial neural networks and other methods if image classification / M.Seetha, I.V. Muralikrishna, Member, IEEE B.L. Deekshatulu, Life Fellow, IEEE, B.L. Malleswari, Nagaratna, P. Hedge - K.: Journal of Theoretical and Applied Information Technology
16. Neural Network a Comprehensive Foundation; a Computational Approach to

Learning and Machine Intelligence / S Haykin, - Macmillan, NY, 1994.

17. The Nature of Statistical Learning Theory / V.N.Vapnik, (New York: Springer-Verlag), 1995.

18. Statistical Learning Theory (New York: Wiley) / V. N Vapnik, 1998.

19. A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery / C. J. C. Burges, 2, 1998, с. 121–167

20. R. K. Sundaram, A First Course in Optimization Theory, (New York: Cambridge University Press), 1996.

21. Introduction to Convolutional Neural Networks / Jianxin Wu. LAMDA Group. National Key Lab for Novel Software Technology. Nanjing University, China, 2017.

22. Преимущества Python перед другими языками программирования [Электронный ресурс]: [Веб-сайт]. Режим доступа: <https://www.digest.pro/news/preimushhestva-python-pered-drugimi-jazykami-programmirovaniya/>, 2019.

23. Динамика популярности языков программирования для использования в машинном обучении [Электронный ресурс]. Режим доступа: <https://trends.google.com/trends/explore?date=today%205-y&q=Python%20Machine%20Learning,R%20Machine%20Learning,Java%20Machine%20Learning,Scala%20Machine%20Learning,Julia%20Machine%20Learning>, 2020.

24. Python documentation [Электронный ресурс]. Режим доступа: <https://www.python.org/doc/>, 2020.

ДОДАТОК А Тексти програмного коду

Файл classifier.py

```

import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten, Dense, Conv2D, MaxPool2D,
Dropout
from sklearn.model_selection import train_test_split
from tkinter import *
import os
import cv2
import csv

CATEGORIES = ["correct", "incorrect"]
#DATA_DIR =
"/home/administrator/KPI/Graduate_work/Program/certificates_dataset"
IMG_SIZE = 20
EPOCHS = 60

class ImageDataset:
    def __init__(self, DATA_DIR):
        self.data = []
        self.img_names = []
        self.DATA_DIR = DATA_DIR

    def create_data(self):
        for category in CATEGORIES:
            path = os.path.join(self.DATA_DIR, category) # path to
certificates
            class_num = CATEGORIES.index(category)
            for img in os.listdir(path):
                try:
                    img_array = cv2.imread(os.path.join(path, img),
cv2.IMREAD_GRAYSCALE)
                    self.img_names.append(img)
                    new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
                    self.data.append([new_array, class_num])
                except Exception as e:
                    pass

    def prepare_data(self):
        X = []
        y = []
        for features, label in self.data:
            X.append(features)
            y.append(label)

        X = np.array(X)
        X = X.reshape(X.shape[0], IMG_SIZE, IMG_SIZE, 1)
        X = X / 255.0

```

					ДП ІС-6212.0000-с.ПЗ	Арк.
						85
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        y = np.array(y)
        indices = np.arange(len(self.data))
        X_train, X_test, y_train, self.y_test, idx1, self.idx2 =
train_test_split(X, y, indices, test_size = 0.3, random_state = 0, stratify =
y)

        return X_train, X_test, y_train, self.y_test, idx1, self.idx2

    def write_to_csv(self, pred):
        with open('bin_classification.csv', 'w', newline='') as file:
            writer = csv.writer(file, delimiter=';')
            writer.writerow(["id", "name", "category", "category_num",
"correct_category_num"])
            for i in range(len(self.idx2)):
                writer.writerow([int(i + 1), self.img_names[self.idx2[i]],
CATEGORIES[int(pred[i])], int(pred[i]), int(self.y_test[i])])

class Model:
    def __init__(self, X_train, y_train, X_test, y_test):
        self.X_train = X_train
        self.y_train = y_train
        self.X_test = X_test
        self.y_test = y_test

    def build_model(self):
        self.model = Sequential()
        self.model.add(Conv2D(filters=32, kernel_size=(3,3), padding='same',
activation='relu', input_shape = [IMG_SIZE, IMG_SIZE, 1]))

        self.model.add(Conv2D(filters=32, kernel_size=(3, 3), padding='same',
activation='relu'))
        self.model.add(MaxPool2D(pool_size=(2, 2), strides=2,
padding='valid'))
        self.model.add(Dropout(0.5))

        self.model.add(Conv2D(filters=32, kernel_size=(3, 3), padding='same',
activation='relu'))
        self.model.add(MaxPool2D(pool_size=(2, 2), strides=2,
padding='valid'))
        self.model.add(Dropout(0.5))

        self.model.add(Flatten())
        self.model.add(Dense(units=128, activation='relu'))
        self.model.add(Dense(1, activation='sigmoid'))
        self.model.summary()
        self.model.compile(optimizer = 'adam', loss = 'binary_crossentropy',
metrics = ['accuracy'])
        self.history = self.model.fit(self.X_train, self.y_train, epochs =
EPOCHS, verbose=1, validation_data=(self.X_test, self.y_test))
        self.model.save("bin_class_model.h5")

    def recognize_class(self):
        self.y_pred = self.model.predict_classes(self.X_test)

    def plot_accuracy(self):

```

```

epoch_range = range(1, EPOCHS+1)
plt.plot(epoch_range, self.history.history['accuracy'])
plt.plot(epoch_range, self.history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc = 'upper left')
plt.show()

class CommandAPI():
    def __init__(self):
        self.BinaryModel = Model([], [], [], [])

    def upload_data(self, DATA_DIR):
        self.Dataset = ImageDataset(DATA_DIR)
        self.Dataset.create_data()

    def train_model(self):
        X_train, X_test, y_train, y_test, idx1, idx2 =
self.Dataset.prepare_data()
        self.BinaryModel = Model(X_train, y_train, X_test, y_test)
        self.BinaryModel.build_model()

    def plot_accuracy(self):
        self.BinaryModel.plot_accuracy()

    def classify_certificates(self):
        self.BinaryModel.recognize_class()

    def write_csv(self):
        self.Dataset.write_to_csv(self.BinaryModel.y_pred)

class Form():
    def __init__(self):
        self.window = Tk()
        self.window.geometry("450x400")
        self.window.title("Classifier")
        self.Command = CommandAPI()

    def display_form(self):
        self.path_data = StringVar()
        labell = Label(self.window, text="Welcome to certificate classifier",
fg='dark blue', font=("arial", 14, "bold"))
        labell.pack(fill=BOTH, pady=20, padx=5, expand=False)
        label_path = Label(self.window, text="Path to image dataset\n Format:
'/C.../certificates_dataset' ", width=50, font=("arial", 12, "bold"))
        label_path.pack()
        entry_path = Entry(self.window, textvar=self.path_data)
        entry_path.pack(pady=5)
        button_load = Button(self.window, text="Upload data",
command=self.set_path, font=("arial", 12))
        button_load.pack(fill=BOTH, padx=50)

```

```

        button1 = Button(self.window, text="1. Train the model", width=12,
command=self.Command.train_model, font=("arial", 12))
        button1.pack(fill=BOTH, padx=50, pady=5)
        button2 = Button(self.window, text="2. Display model accuracy",
width=12, command=self.Command.plot_accuracy, font=("arial", 12))
        button2.pack(fill=BOTH, padx=50)
        button3 = Button(self.window, text="3. Classificate certificates on
correct/incorrect", width=12, command=self.Command.classify_certificates,
font=("arial", 12))
        button3.pack(fill=BOTH, padx=50, pady=5)
        button4 = Button(self.window, text="4. Write results in csv-file",
width=12, command=self.Command.write_csv, font=("arial", 12))
        button4.pack(fill=BOTH, padx=50)
        self.window.mainloop()

    def set_path(self):
        self.DATA_DIR=self.path_data.get()
        if self.DATA_DIR:
            self.Command.upload_data(self.DATA_DIR)
        else:
            print("Empty path")

if __name__ == "__main__":
    form = Form()
    form.display_form()

```

Файл mark.py

```

import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten, Dense, Conv2D, MaxPool2D,
Dropout
from sklearn.metrics import accuracy_score, confusion_matrix
from mlxtend.plotting import plot_confusion_matrix
from sklearn.model_selection import train_test_split
from pprint import pprint
import os
import cv2
import csv

DATA_DIR = "/home/administrator/KPI/Graduate_work/Program/marks_dataset"
CATEGORIES = ["four", "five", "six", "seven", "eight", "nine", "ten",
"eleven", "twelve"]
IMG_SIZE = 36
EPOCHS = 60

def create_data():
    mark_names = []

```

					ДП IC-6212.0000-с.ПЗ	Арк.
						88
ЗМН.	Арк.	№ докум.	Підпис	Дата		


```

data = []

for category in CATEGORIES:
    path = os.path.join(DATA_DIR, category)      # path to marks
    class_num = CATEGORIES.index(category)
    for img in os.listdir(path):
        try:
            img_array = cv2.imread(os.path.join(path, img),
cv2.IMREAD_GRAYSCALE)
            mark_names.append(img)
            new_array = cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
            data.append([new_array, class_num])
        except Exception as e:
            pass
    return data, mark_names

def prepare_data(data, mark_names):
    X = []
    y = []
    for features, label in data:
        X.append(features)
        y.append(label)

    X = np.array(X)
    X = X.reshape(X.shape[0], IMG_SIZE, IMG_SIZE, 1)
    X = X / 255.0
    y = np.array(y)
    indices = np.arange(len(data))
    print(len(X))
    X_train, X_test, y_train, y_test, idx1, idx2 = train_test_split(X, y,
indices, test_size=0.4, random_state=0, stratify=y)
    return X_train, X_test, y_train, y_test, idx1, idx2

def build_model(X_train, y_train, X_test, y_test):
    model = Sequential()
    model.add(Conv2D(filters=32, kernel_size=(3,3), padding='same',
activation='relu', input_shape = [IMG_SIZE, IMG_SIZE, 1]))

    model.add(Conv2D(filters=32, kernel_size=(3, 3), padding='same',
activation='relu'))
    model.add(MaxPool2D(pool_size=(2, 2), strides=2, padding='valid'))
    model.add(Dropout(0.5))

    model.add(Conv2D(filters=32, kernel_size=(3, 3), padding='same',
activation='relu'))
    model.add(MaxPool2D(pool_size=(2, 2), strides=2, padding='valid'))
    model.add(Dropout(0.5))

    model.add(Flatten())
    model.add(Dense(units=128, activation='relu'))
    model.add(Dense(units=len(CATEGORIES), activation='softmax'))
    model.summary()
    model.compile(optimizer = 'adam', loss =

```

```

'sparse_categorical_crossentropy', metrics = ['accuracy'])
    history = model.fit(X_train, y_train, epochs = EPOCHS, verbose=1,
validation_data=(X_test, y_test))

    return history, model

def recognize_marks(model, X_test, y_test=[]):
    y_pred = model.predict_classes(X_test)
    return y_pred

def plot_accuracy(history):
    epoch_range = range(1, EPOCHS+1)
    plt.plot(epoch_range, history.history['accuracy'])
    plt.plot(epoch_range, history.history['val_accuracy'])
    plt.title('Model accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Val'], loc = 'upper left')
    plt.show()

def write_to_csv(names, pred, y_test, idx2):
    with open('mark_classification.csv', 'w', newline='') as file:
        writer = csv.writer(file, delimiter=';')
        writer.writerow(["id", "name", "pred_mark", "mark"])
        for i in range(len(idx2)):
            writer.writerow([int(i+1), names[idx2[i]], int(pred[i]) + 4,
int(y_test[i]) + 4])

if __name__ == '__main__':
    data, mark_names = create_data()
    X_train, X_test, y_train, y_test, idx1, idx2 = prepare_data(data,
mark_names)

    history, model = build_model(X_train, y_train, X_test, y_test)
    y_pred = recognize_marks(model, X_test)
    plot_accuracy(history)
    conf_mat = confusion_matrix(y_test, y_pred)
    plot_confusion_matrix(conf_mat, figsize=(len(CATEGORIES),
len(CATEGORIES)), class_names=CATEGORIES)
    plt.show()
    write_to_csv(mark_names, y_pred, y_test, idx2)

```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Кафедра автоматизованих систем обробки інформації і управління

УЗГОДЖЕНО

Керівник проєкту

(підпис) О.Д. Фіногенов
(ініціали, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

(підпис) О.А. Павлов
(ініціали, прізвище)

“14” квітня 2020 р.

Інформаційна система розпізнавання тексту технології класифікації
додатків до атестатів

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр ДП ІС-6212.01.000 ТЗ

на 9 сторінках

Київ – 2020 року

Зміст

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	93
1.1 Повне найменування системи та її умовне позначення	93
1.2 Найменування організації-замовника та організацій-учасників робіт	93
1.3 Перелік документів, на підставі яких створюється система (Завдання на ДП)	93
1.4 Планові терміни початку і закінчення роботи зі створення системи	93
2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ	94
2.1 Призначення системи	94
2.2 Цілі створення системи	94
3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ	95
4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	96
4.1 Вимоги до функціональних характеристик	96
4.2 Вимоги до надійності	96
4.3 Вимоги до складу і параметрів технічних засобів	97
5 СТАДІЇ І ЕТАПИ РОЗРОБКИ	98
6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ	99
6.1 Види випробувань	99

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Найменування системи: Інформаційна система розпізнавання тексту технології класифікації додатків до атестатів.

Умовне позначення: Text image classification

1.2 Найменування організації-замовника та організацій-учасників робіт

Найменування організації замовника: Кафедра АСОІУ, ФІОТ, НТУУ «КПІ ім. Ігоря Сікорського»;

Організація-учасник робіт: студент ФІОТ, групи ІС-62, Марченков І.Д.

1.3 Перелік документів, на підставі яких створюється система (Завдання на ДП)

При розробці системи і створення проектно-експлуатаційної документації Виконавець повинен керуватися вимогами наступних нормативних документів:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем.

1.4 Планові терміни початку і закінчення роботи зі створення системи

Плановий термін початку: 10 лютого 2020 року;

Плановий термін закінчення: 31 травня 2020 року.

					ДП ІС-62 12.01.000 ТЗ	Арк.
						93
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1 Призначення системи

Система призначена для автоматизації підрахунку середнього балу додатку до атестата (з 2020 року – сертифікат) про здобуття повної загальної середньої освіти вступника під час проведення вступної кампанії. Середній бал атестату є однією зі складових формування рейтингового балу абітурієнта. Під час реєстрації в електронному кабінету вступник має обчислити середній бал додатку самостійно за 12-бальною шкалою з округленням до десятих частин балу та завантажити скан-копію додатку. Приймальна комісія ВНЗ повинна перевірити правильність підрахунку, користуючись скан-копіями.

2.2 Цілі створення системи

- Автоматизувати процес підрахунку середнього балу додатку до атестату вступника;
- Мінімізувати ймовірність виникнення помилки під час обрахунку балу.

					ДП ІС-6212.01.000 ТЗ	Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Об'єктом автоматизації є процеси по управлінню вступною кампанією, а також контроль коректності виконання вказаних процесів, а саме перевірки достовірності підрахунку середнього балу додатку до атестата.

Процеси управління підрахунку балу включають у себе:

- Завантаження бази даних;
- Попередня обробка зображень;
- Класифікація зображень атестатів та зображень оцінок;
- Збереження результатів до бази.

Дані процеси здійснюються наступними спеціалістами:

- Робітник приймальної комісії.

Користувач має можливість вказати адресу директорії із зображеннями додатків, виконати класифікацію, а також натренувати модель на індивідуальному наборі даних із визначеною структурою.

					ДП ІС-6212.01.000 ТЗ	Арк.
						95
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

Задачі, які підлягають автоматизації:

- класифікація зображень за коректністю;
- повертання зображення у правильне положення;
- класифікація зображень рукописних оцінок.

Форма представлення вихідної інформації:

- текстовий файл формату .csv;
- статистичний графік виконаних функцій;
- розбивання класифікованих даних по папкам за приналежністю до класу.

Характеристики точності: точність кожної моделі досягає 90 – 95%.

4.2 Вимоги до надійності

У випадку виникнення непередбачених критичних ситуацій, аварій, відмов, технічних засобів (у тому числі зникнення напруги), збоїв у роботі загальносистемного програмного забезпечення ІС повинна мати можливість відтворення своєї працездатності з резервний копій за короткий проміжок часу та з мінімальними втратами інформації.

ІС повинна поєднувати надійність та функціональність. У разі виникнення аварійних ситуацій необхідно сповіщати користувача та надавати інструкцію для подальших дій. Будь-які аварійні ситуації мають бути задокументовані у звіті, який при необхідності надсилається розробнику для визначення причини збою в роботі та усуненні помилок, які могли привести до нестабільної роботи програмного продукту.

					ДП ІС-6212.01.000 ТЗ	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3 Вимоги до складу і параметрів технічних засобів

Для нормального функціонування ІС потрібен комп'ютер з такими характеристиками:

- процесор з тактовою частотою не нижче 1 ГГц;
- оперативний запам'ятовуючий пристій – 1024 Мб;
- 20 Гб вільного місця на жорсткому диску.

					ДП ІС-6212.01.000 ТЗ	Арк.
						97
Змн.	Арк.	№ докум.	Підпис	Дата		

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Основні етапи розробки ІС.

№ з/п	Назва етапу роботи	Термін виконання етапу	Результат виконання
1	Аналіз вимог	10.02.2020	+
2	Побудова алгоритму	24.02.2020	+
3	Опис предметного середовища	23.03.2020	+
4	Створення бази зображень для тренування нейронної мережі	20.04.2020	+
5	Розробка програмного забезпечення	27.04.2020	+
6	Розробка ТЗ	04.05.2020	+
7	Опис функціональної моделі	11.05.2020	+
8	Огляд наявних аналогів	11.05.2020	+
9	Налагодження програми	18.05.2020	+
10	Тестування програми	25.05.2020	+
11	Захист роботи	15.06.2020	

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ

6.1 Види випробувань

- Тестування нейронної мережі при різній кількості вузлів;
- Тестування нейронної мережі при різному відношенні тренувальних і тестувальних даних;
- Тестування нейронної мережі при різній якості зображень;
- Тестування нейронної мережі при різній кількості епох;
- Тестування швидкодії.

					ДП ІС-6212.01.000 ТЗ	Арк.
						99
Змн.	Арк.	№ докум.	Підпис	Дата		

Власник документу:
Попенко Володимир Дмитрович
Дата перевірки:
11.06.2020 01:36:51 EEST
Дата звіту:
12.06.2020 01:26:03 EEST

ID перевірки:
1003947938
Тип перевірки:
Doc vs Internet + Library
ID користувача:
77149

Назва документу: Marchenkov_is62

ID файлу: 1003963100 Кількість сторінок: 85

Кількість слів: 12852 Кількість символів: 92483 Розмір файлу: 8.51 MB

13.6% Схожість

Найбільша схожість: 1.46% з джерело <https://datascience.stackexchange.com/q/48390>

9.24% Схожість з Інтернет джерелами

30

Page 87

9.86% Текстові збіги по Бібліотеці акаунту

45

Page 93

0.45% Цитат

Цитати

5

Page 94

Вилучення переліку посилань вимкнено

0% Вилучень

Вилучений текст відсутній

Підміна символів

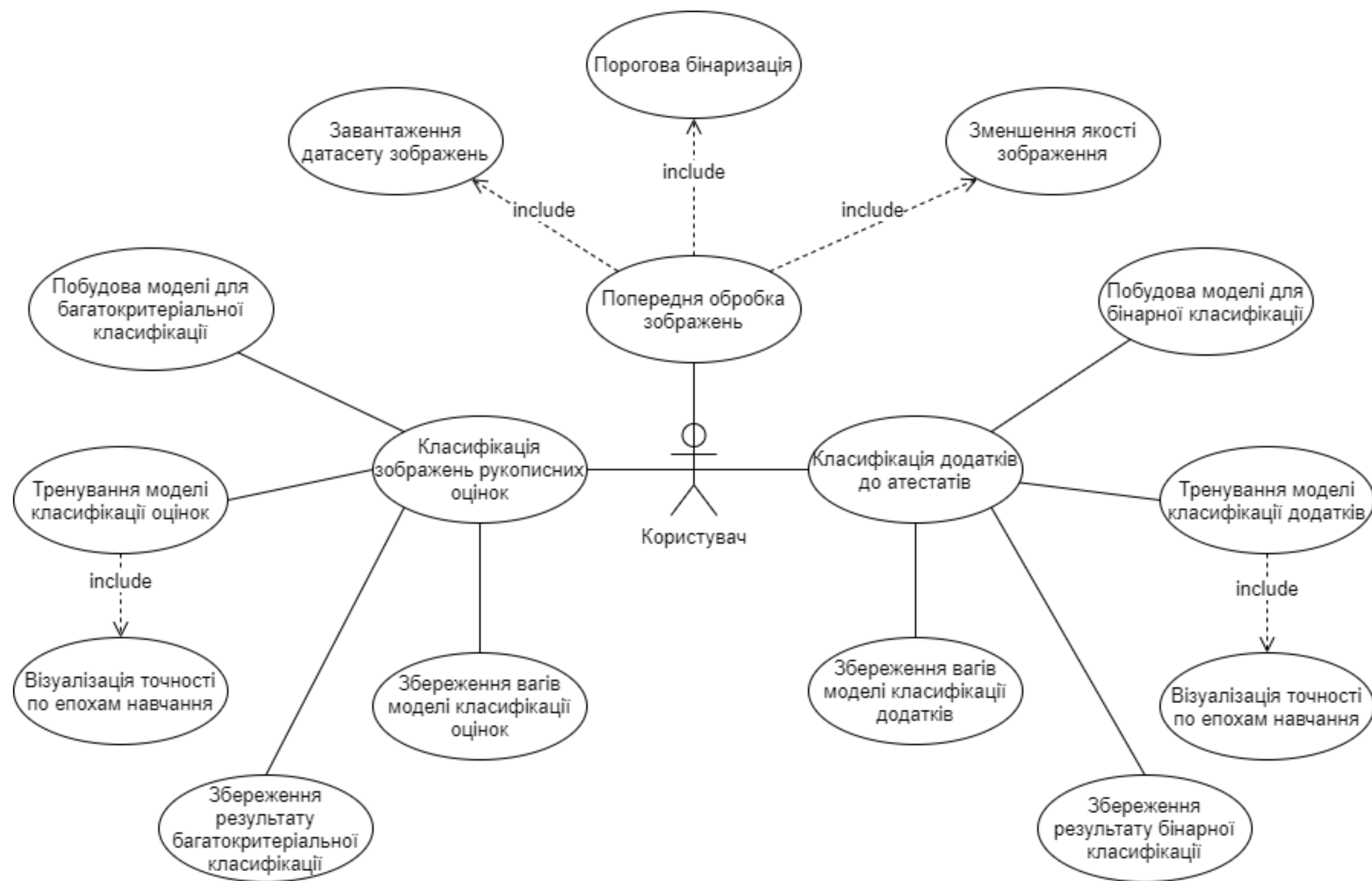
Заміна символів

10

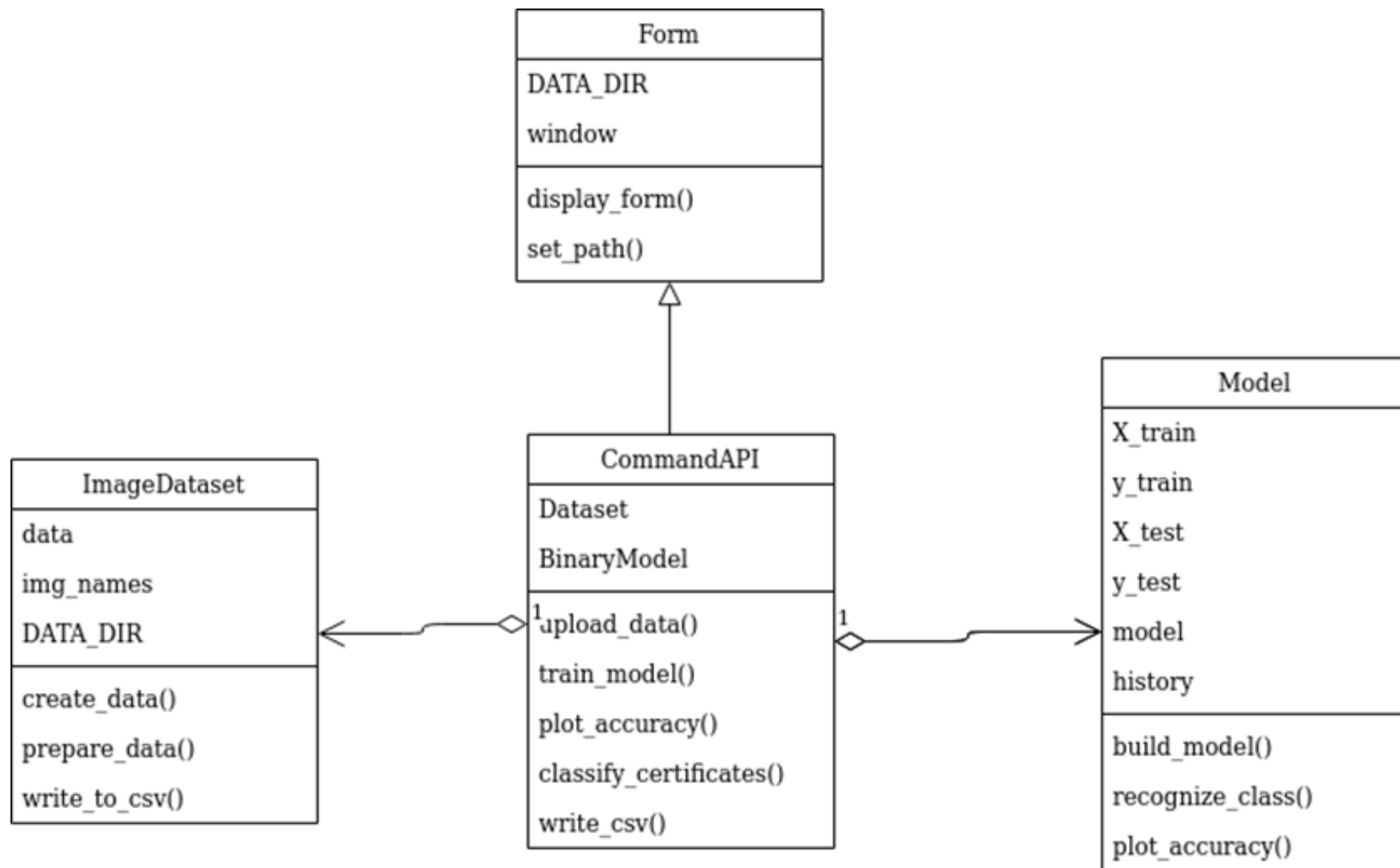
Графічний матеріал до дипломного проєкту

на тему: Інформаційна система розпізнавання тексту технології
класифікації додатків до атестатів

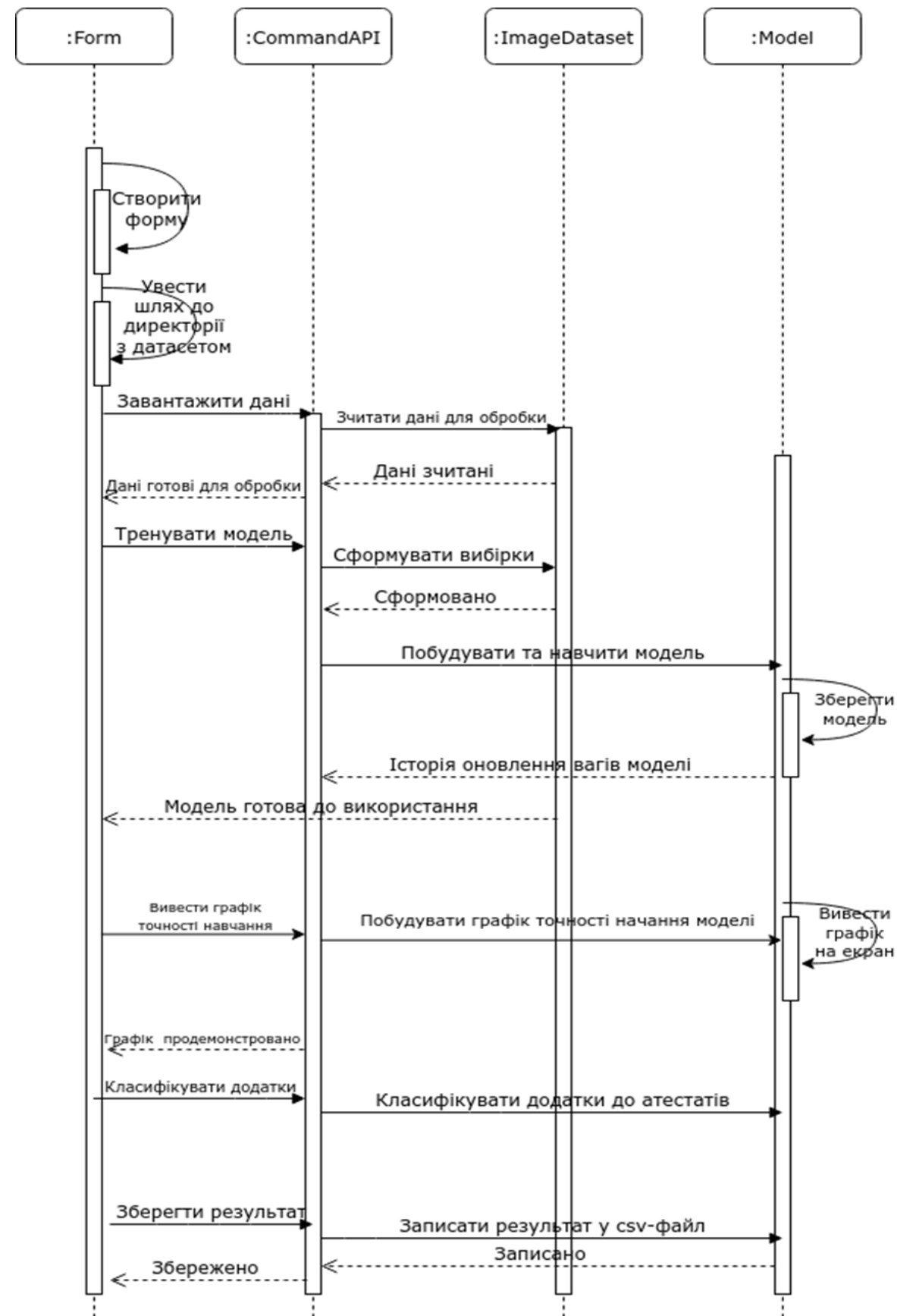
Київ – 2020 року



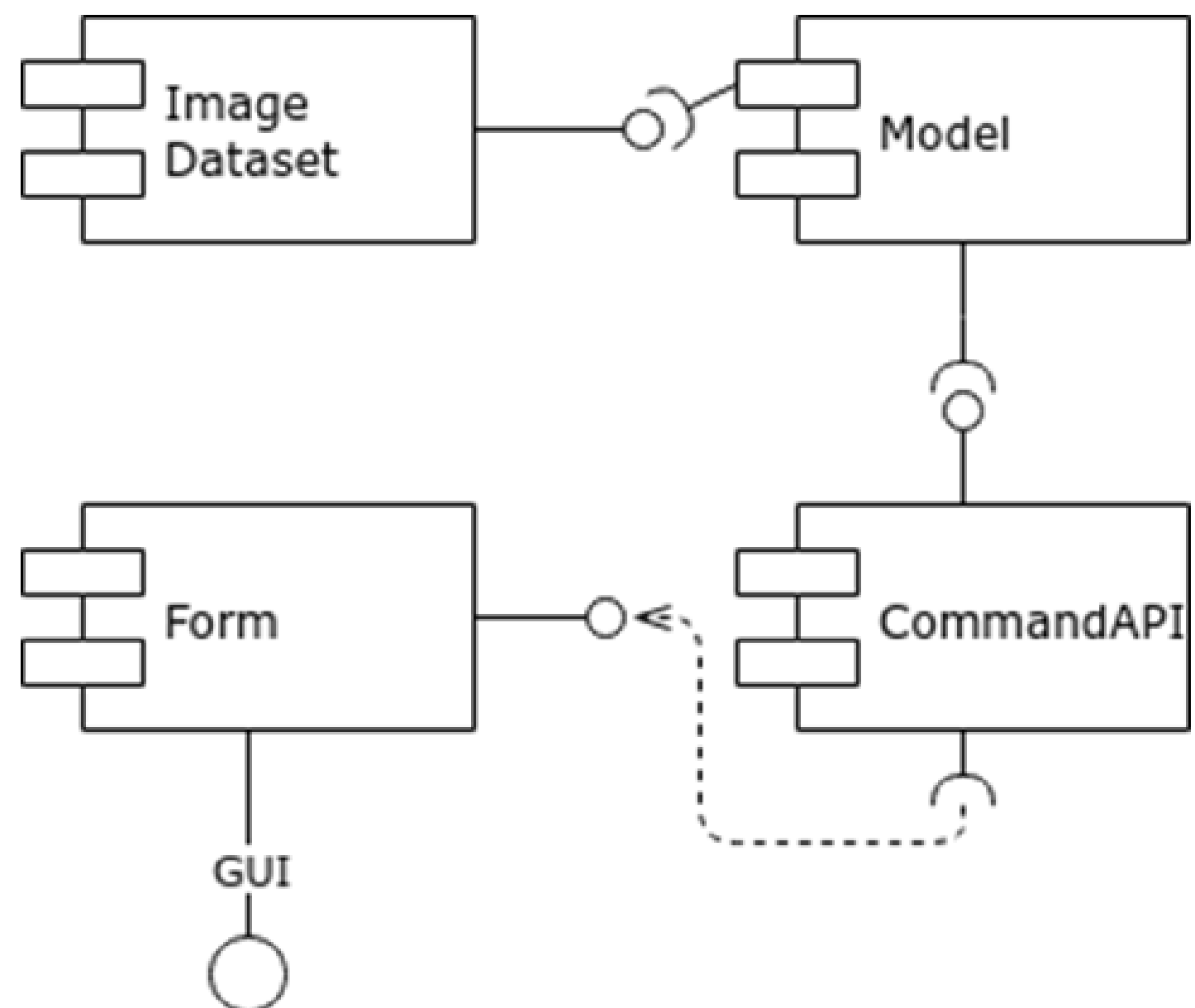
					ДП 6212.02.000 ССВ			
Зм.	Арк.	ПІБ	Підп.	Дата	Схема структурна варіантів використання	Літ.	Аркуш	Аркуші
Розробн.		Марченко І.Д.					1	1
Керівн.		Фіногенов О.Д.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		
Консульт.								
Н/контр.		Новінський В.П.						
Затв.		Фіногенов О.Д.						



					ДП 6212.03.000 ССК			
Зм.	Арк.	ПІБ	Підп.	Дата	Схема структурна класів програмного забезпечення	Літ.	Аркуш	Аркушів
Розробн.		Марченко І.Д.					1	1
Керівн.		Фіногенов О.Д.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		
Консульт.								
Н/контр.		Новінський В.П.						
Затв.		Фіногенов О.Д.						



					ДП 6212.04.000 ССД			
Зм.	Арк.	ПІБ	Підп.	Дата	Схема структурна діяльності			
Розробн.		Марченко І.Д.						
Керівн.		Фіногенов О.Д.						
Консульт.								
Н/контр.		Новінський В.П.						
Затв.		Фіногенов О.Д.						
					Літ.			Аркуш
								Аркуші
								1
								1
					КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62			



					ДП 6212.05.000 ССК			
Зм.	Арк.	ПІБ	Підп.	Дата	Схема структурна компонентів програмного забезпечення	Літ.	Аркуш	Аркушів
Розробн.		Марченко І.Д.					1	1
Керівн.		Фіногенов О.Д.				КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62		
Консульт.								
Н/контр.		Новінський В.П.						
Затв.		Фіногенов О.Д.						

Бінарний класифікатор

Класифікатор додатків до атестатів

Шлях до директорії з даними
Формат: YC.../certificates_dataset'

Завантажити датасет

1. Тренувати модель

2. Продемонструвати точність моделі

3. Класифікувати додатки на коректні/некоректні

4. Записати результат у csv-файл

Терминал

Файл Правка Вид Поиск Терминал Вкладки Справка

administrator@administ... x Терминал x Терминал x

Layer (type)

Output Shape

Param #

=====

conv2d (Conv2D)

(None, 20, 20, 32)

320

conv2d_1 (Conv2D)

(None, 20, 20, 32)

9248

max_pooling2d (MaxPooling2D)

(None, 10, 10, 32)

0

dropout (Dropout)

(None, 10, 10, 32)

0

conv2d_2 (Conv2D)

(None, 10, 10, 32)

9248

max_pooling2d_1 (MaxPooling2D)

(None, 5, 5, 32)

0

dropout_1 (Dropout)

(None, 5, 5, 32)

0

flatten (Flatten)

(None, 800)

0

dense (Dense)

(None, 128)

102528

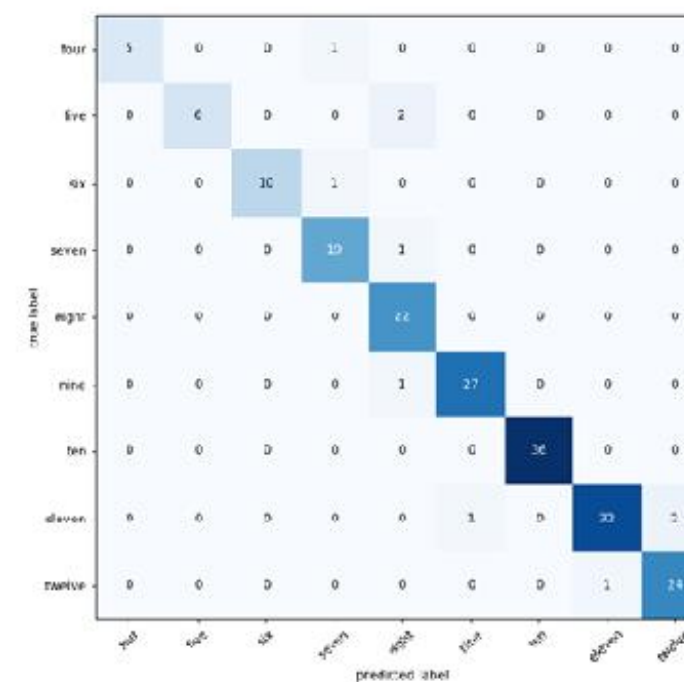
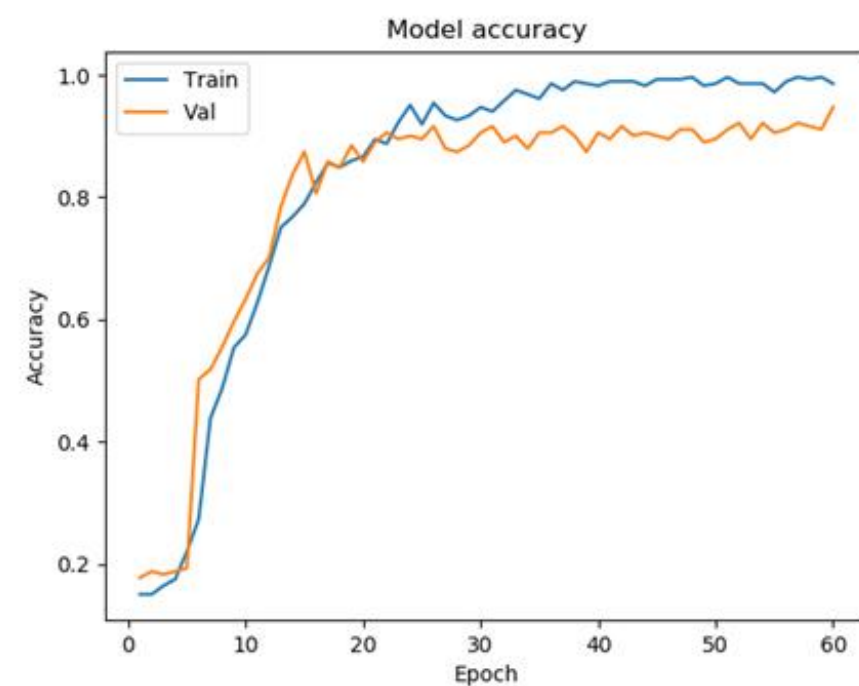
dense_1 (Dense)

(None, 1)

129

=====

Total params: 121,473



					ДП 6212.06.000 КЕ				
Зм.	Арк.	ПІБ	Підп.	Дата	Креслення вигляду екранних форм	Літ.	Аркуш	Аркушів	
Розробн.		Марченко І.Д.							
Керівн.		Фіногенов О.Д.					1	1	
Консульт.						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-62			
Н/контр.		Новінський В.П.							
Затв.		Фіногенов О.Д.							

id	name	category	category_num	correct_category_num
1	Atestat_3711285.jpg	correct	0	0
2	Atestat_3708986.jpg	correct	0	0
3	Atestat_4060475.jpg	correct	0	0
4	Atestat_3711573.jpg	correct	0	0
5	Atestat_3711688.jpg	incorrect	1	0
6	Atestat_4043203.jpg	incorrect	1	1
7	Atestat_3708646.jpg	correct	0	0
8	Atestat_4258280.jpg	correct	0	0
9	Atestat_3711223.jpg	correct	0	0
10	Atestat_3711336.jpg	correct	0	0
11	Atestat_3711180.jpg	correct	0	0
12	Atestat_4258586.jpg	correct	0	0
13	Atestat_3712914.jpg	correct	0	0
14	Atestat_4177395.jpg	incorrect	1	1
15	Atestat_4272694.jpg	incorrect	1	1
16	Atestat_3712195.jpg	correct	0	0
17	Atestat_3707599.jpg	correct	0	0
18	Atestat_3711816.jpg	correct	0	0
19	Atestat_3707814.jpg	correct	0	0
20	Atestat_3707443.jpg	correct	0	0
21	Atestat_3984851.jpg	incorrect	1	1
22	Atestat_4259510.jpg	correct	0	0
23	Atestat_4060219.jpg	correct	0	0
24	Atestat_3708221.jpg	correct	0	0
25	Atestat_3708527.jpg	correct	0	0
26	Atestat_4258436.jpg	correct	0	0

id	name	pred_mark	mark
1	mark_3956848_5.png	8	5
2	mark_3952667_8.jpg	8	8
3	mark_3935665_11.jpg	11	11
4	mark_3940156_6.jpg	6	6
5	mark_3951340_11.jpg	11	11
6	mark_3957933_4.jpg	4	4
7	mark_3828625_8.jpg	6	8
8	mark_3706504_10.jpg	10	10
9	mark_3735827_10.jpg	10	10
10	mark_3829210_6.jpg	6	6
11	mark_3952540_8.jpg	7	8
12	mark_3826932_11.jpg	11	11
13	mark_3825072_12.jpg	11	12
14	mark_3954116_7.jpg	7	7
15	mark_3939943_10.jpg	10	10
16	mark_3956700_5.jpg	5	5
17	mark_3948119_8.jpg	8	8
18	mark_3941236_12.jpg	11	12
19	mark_4188929_11.jpg	11	11
20	mark_3947577_9.jpg	9	9
21	mark_4188929_7.jpg	7	7
22	mark_3829938_7.jpg	7	7
23	mark_3949592_8.jpg	8	8
24	mark_3953803_10.jpg	10	10
25	mark_3949972_12.jpg	12	12
26	mark_3940836_10.jpg	10	10

					ДП 6212.07.000 КЗ			
Зм.	Арк.	ПІБ	Підп.	Дата				
Розробн.		Марченко І.Д.			Креслення вигляду звітних форм		Літ.	Аркуш
Керівн.		Фіногенов О.Д.						1
Консульт.							КПІ ім. Ігоря Сікорського	
Н/контр.		Новінський В.П.					Каф. АСОІУ	
Затв.		Фіногенов О.Д.					Гр. ІС-62	